An Encoder-Decoder based Deep Learning Approach for Anonymization of Visual Surveillance Media with Preservation of Utility

EIVIND LINDSETH

SUPPERVISORS

Morten Goodwin, Lei Jiao, Bente Skattør

University of Agder, 2020
Faculty of Engineering and Science
Department of ICT

Master

Abstract

The field of computer vision has seen significant progress recently following innovations in deep learning neural networks. Activity can be identified from surveillance cameras. Automatic detection of unwanted incidents would enable police to act quickly with appropriate resources. Activity detecting machine learning algorithms need many examples in its learning phase. However, videos from surveillance cameras may contain privacy-sensitive information. The videos are not allowed to be used outside of the police unless anonymized. However, traditional anonymization techniques remove visual information, reducing utility as training data.

This thesis introduces a method for anonymization of visual surveillance media, with preservation of utility. A face is anonymized by applying changes to many facial attributes using a novel encoder-decoder face editing network. This results in a natural-looking non-existing face that has a high chance of being detected. The editing process is controlled by attributes, but it is not known which combinations are suitable. The selection process is explored, and it is shown that attributes for best performance must be set individually per face. The amount of change applied is measured using face embedding distance. We measure the anonymization rate using a reverse image search and learn the distance number to achieve anonymization. By proper selection of editing attributes it possible to achieve both high anonymization and high face detection. This thesis did not attempt to find the optimum attribute model. Other problems discovered needs to be resolved also.

A missed detection in video results in lost anonymization. This thesis also proposes a short term detection memory to the Darknet object detector framework. This memory preserves detections over a number of video frames. Experiments show improved recall, but without object tracking, it introduces false positives on moving objects.

Preface

This thesis concludes the master's education in Communication and Information Technology (ICT), at the University of Agder, Norway. "Non-Intrusive Surveillance" started in 2018 as a possibility study of using AI in the Norwegian police force. It is a concept for use of video, images, sound and sensor data. In this thesis, we research the problem of anonymizing video material from surveillance cameras, that will be further used to train systems for detecting unwanted incidents. The police will use partners for crowd sourcing / open innovation. But data has to be anonymized before it can be shared with partners.

Several people have supported and contributed to the completion of this project. I want to thank in Dr. Bente Skattør, Oslo Police District, for the project and good discussions. Associate Professor Morten Goodwin and Dr. Lei Jiao for guidance througout the thesis, and Postdoctoral fellow Vimala Nunavath for her contributions to finishing the report.

Table of Contents

Abstract iii

Preface v

Glossary xi

List of Figures xiii

List of Tables xv

1 Introduction 1

1.1 Motivation 2

1.2 Problem Statement 2

1.3 Research questions 3

1.4 Thesis goals 4

1.5 Assumptions and Limitations 4

1.5.1 Assumptions 4

1.5.2 Limitations 4

1.6 Contributions 5

1.7 Thesis outline 5

2 Theoretical background 7

2.1 Background 7

2.1.1 Privacy and Utility 7

2.1.2 Privacy protection methods 8

2.1.3 Evaluation of privacy techniques 11

2.1.4 Object detection 12

2.1.5 Object detection accuracy 20

2.1.6 Generative Models 20

2.1.7 Baseline 22

2.2 State-of-the-art 24

3 Research Methodology 27
3.1 Privacy filter 27
3.1.1 Dataset and data-preprocesing 28
3.1.2 Implementation 28
3.1.3 Attribute Generator 30
3.1.4 Privacy sensitive region 30
3.1.5 Preprocessing filter 31
3.1.6 Attribute learning 32
3.2 Evaluation methods 33
3.2.1 Privacy evaluation 33
3.2.2 Utility evaluation 35
3.3 Detection short term memory 36
3.3.1 Functional description 36
3.3.2 Datasets 37
3.3.3 Detection identification 37

4 Experiments and results 39
4.1 Experiment1: Privacy filter 39
4.1.1 Anonymization and face detection 40
4.1.2 Distance vs anonymization 45
4.1.3 Pre-processing filter 46
4.2 Experiment2: Accuracy 47
4.3 Experiment3: Learning attributes 49
4.3.1 Predict visual appearance 50
4.3.2 Testing the model on different faces 52
4.3.3 Maximizing a utility function 52
4.4 Experiment4: Detection Memory 54

5 Conclusion and Future Work 59
5.1 Conclusion 59
5.2 Future Work 61

References 67

Appendices 69
A Hardware Specification 69
B Anonymization 69
C Object detection 70
C.1 Annotate video 70
C.2 Short Term Detection Memory 70

C.3 Blurring anonymization filter 71

Glossary

AttGAN Facial Attribute Editing GAN. 22

CNN Convolutional Neural Network. 14

GAN Generative Adversarial Networks. 3

IoT Internet of Things. 1

IoU Intersection over Union. 20

MTCNN Multi-Task Cascaded Convolutional Neural Network. 18

NN Neural Network. 13

STDM Short Term Detection Memory. 36

TP True Positive. 20

YOLO You only Look Once object detector. 16

List of Figures

2.1 CNN image classification pipeline. 15
2.2 YOLO object classification. 17
2.3 MTCNN pipeline. 19
2.4 Blur image filter. 23
2.5 Pixelating image filter. 23
2.6 Anonymization errors vs face detections. 23

3.1 Multiple anonymization results. 30
3.2 Preprocessing by pixelating. 32
3.3 Distance measurement region 34
3.4 Distance matrix. 35
3.5 Collision detection. 38

4.1 Faces used for anonymization. 41
4.2 Anonymization using random attributes. 41
4.3 Anonymization success rate through the ten batches. 43
4.4 Distance and anonymization 44
4.5 Pre-filter with pixelation. 47
4.6 Sideways head poor performance. 49
4.7 Attributes vs distance plot. 50
4.8 Classification examples of bad (top) and good (bottom). . . 51
4.9 Detection restored from memory. 54
4.10 Face detection on anonymized video. 56
4.11 Measuring effect of STDM. 57
4.12 mAP and recall vs memory length. 58

1 VGGFaces2 examples: Dalai Lama 69
2 WIDER Faces example anonymized using new method 70

List of Tables

3.1 Distance distribution VGG faces. 35

4.1 Distance distribution face id 6011. 44
4.2 Sampling anonymization in mini-batches of 30 grouped on distance. 46
4.3 Face identification count using pre-processing. 47
4.4 Accuracy measurements. 48
4.5 Learning parameters. 50
4.6 Maximizing utility function. 53

1 Hardware specification 69

Chapter 1

Introduction

There has been an increase in privacy concerns, as the amount of sensor data from many different sources is growing fast. Surveillance cameras are abundant, used in streets and public spaces, public transport, shopping malls, and even for home security. Surveillance camera density has grown tremendously, especially in Asia. London is one of two western cities on the top 10 list. Gartner predicts that 5G IoT installed endpoints for outdoor surveillance cameras will reach 11.2 million units in 2022 [12]. This volume should raise concerns about the privacy of the individuals.

Policing in smart cities is a concept where multiple types of sensors help the police to act early. Video cameras have been used for surveillance for a long time, and the numbers are rapidly increasing. But monitoring all the surveillance cameras using human operators is impossible because of the amount of data, both in terms of the numbers and the complex information in the videos, Sulman et al. [39]. Instead, machine learning algorithms can be trained to detect unwanted incidents, using videos of real events as examples. The video streams can then be monitored in real-time. The training process will be handled by the police, but also by external partners to the police. This requires the removal of privacy-sensitive information. This has lead to a need for improved anonymization methods that balance privacy protection and preservation of visual information. Anonymization must preserve as much visual information as possible, to maintain the usefulness of the video as training data.

Recent innovations in machine learning such as Generative Adversarial Networks (GAN) brings new possibilities to generate and edit face images.

1.1 Motivation

Video, images, and sound are some of the data sources (sensors) for the concept “Non-Intrusive Surveillance” by the Oslo Police District. The data will be subjected to pattern recognition for unwanted incidents using machine learning, through open innovation. This has resulted in the need for new anonymization methods. All data, including video and images must be anonymized before published for open innovation. The goal is to train action detection algorithms to detect unwanted incidents in real-time. This can trigger an alarm for manual inspection of the video, or notify police nearby. Unwanted incidents are, for example, violence in the streets, tagging, burglary, and even fallen persons. Training action detection systems require example videos with the relevant actions. However, privacy-sensitive information must be removed before sharing for open innovation. This thesis focuses on the problem of anonymizing faces while maintaining the visual information. It is important that an anonymized face still can be detected. Visual information such pose and gender can also be valuable information, and should be preserved.

1.2 Problem Statement

The problem is how persons can be anonymized in such a way that the anonymized data is highly useful for a future action detection machine learning scenario. The anonymized data will be exposed to various types of pattern recognition, the exact types are unknown at present. By preserving a natural-looking face, its pose, or the heads direction, and gender, enough intelligibility of the video will be preserved. Traditional anonymization methods such as image blurring, pixelation, and masking remove valuable information. This degrades the usefulness of the data. Face detection rate decreases, and this directly reduces the utility of the data. This thesis aims to preserve certain visual information from the face, to allow for different future activity recognition algorithms.

Recent innovations in deep learning-based generative adversarial networks, GAN have opened up for new anonymization methods. This thesis proposes a generic face anonymization method that aims to change the face so much that face recognition algorithms fail, while at the same time face detection algorithms will succeed. It also tries to maintain a visually appealing result, to maintain the data's usability as much as possible.

This thesis is also concerned with improving object detection rate. Objects to be anonymized must first be detected. A missed detection means a missed anonymization. It is therefore important to maximise the performance of the object detector.

1.3 Research questions

The thesis makes an effort to answer the following research questions

1. *Is it possible to achieve both face anonymization and preserve visual information at the same time?*
   To answer this research question, we create new privacy based on a state-of-the-art face attribute editor. This will change rather than of replacing or filter the face.
2. *When is a face modified enough to be classified as anonymized?* A numerical value that represents the change can indicate the level of anonymization. There may exist a threshold value where anonymization is sufficient.
3. *How to control the editing process to achieve good results?*
   To answer this question, labeled datasets can be created. By mapping the parameters, or attributes, to the results, a machine learning algorithm may learn the mapping function.
4. *Can anonymization be improved by adding memory to the object detector?*
   The YOLO object detector analyzes each video frame independently. Can adding a detection memory that keeps information between frames increase detection recall?

1.4 Thesis goals

1. Examine the state of the art of face modification frameworks. We want to modify existing face image in order to preserve important visual information. Build a privacy filter prototype based on face editing.
2. Find suitable performance parameters to measure anonymization efficacy to answer research question 2.
3. Perform experiments to learn the strengths and weaknesses of the privacy filter. Identify what needs to be improved. Create a dataset which maps attributes to outcome. Machine learning algorithms can be used to predict, and improve the results.
4. Extend an existing object detector framework with a memory that keeps detections between frames. The idea is that if the object detector fails to detect an object occasionally, the detection can be restored from memory. The object detector YOLO and its Darknet framework will be used.

1.5 Assumptions and Limitations

1.5.1 Assumptions

1. The privacy filter shall preserve gender. This requires the knowledge of the gender of the person in image / video. Such a gender detector is assumed to exist, and is not implemented in the experiments.

1.5.2 Limitations

1. Simple horizontal and vertical alignment was employed to align detected faces as a pre-process step for the AttGAN based privacy filter. It did not compensate for rotation, and this leads to poor result of non-constrained images.
2. Training GAN's on image datasets is very processing intensive. Only the pre-trained models were used for AttGAN.
3. We do not have ground truth annotated video. This is necessary in

order to measure accuracy of object detection. Instead, sample videos were prepared manually, limiting both the amount of video material.

1.6 Contributions

This thesis contributes with a novel concept for anonymization with preservation of utility. It uses an encoder-decoder based network to edit faces in the latent space rather than pixel space. This allows for both natural visual appearance and a high degree of anonymization. Compared to traditional anonymization methods such as blurring, pixelation, we can achieve high anonymization and high face detection rate at the same time. Also, other visual information such as pose and gender is preserved. However, choosing parameters leading to a successful result is not understood well and must be learned. We also discover weaknesses that must be addressed. These are summarized in chapter 5.

A short term detection memory is added to the Darknet framework, which is used by YOLO object detector. This increases the effective recall, by preserving detections in memory. An object that fails to be detected, will also fail to be anonymized. By storing detection in memory, it can be restored if regular detection should fail.

1.7 Thesis outline

The rest of the thesis is structured as follows:

- **Chapter 2** presents relevant background theory and state of the art.
- **Chapter 3** describes the methodology.
- **Chapter 4** presents the experiments and results.
- **Chapter 5** concludes the thesis and presents future work.

Chapter 2

Theoretical background

This chapter starts with describing background on privacy and utility, and also presents different methods to achieve privacy. Machine learning algorithms for object detection, face recognition and generative algorithms are also described. Further, the state of the art section presents the existing literature on recent algorithms which balance anonymization and utility.

2.1 Background

2.1.1 Privacy and Utility

The usefulness of anonymized videos as a dataset will to some degree be less than its original counterpart. This is because some information has been lost, thereby excluding certain operations on the data. Some terms are much used in this thesis and needs an introduction. These terms are:

Privacy protection / anonymization:

Privacy protection consists of preventing information an individual wants to keep private to be exposed to the public [31]. Both "privacy protection" and "anonymization" are found in the literature, both terms meaning the same thing. For visual data, as video and images, privacy protection is the task of concealing the true identity of persons in the visual data. Some methods provides stronger protection, but, as we shall see, also degrades the utility of the dataset.

De-anonymization:

The act of revealing the true identities from anonymized data. Sometimes the anonymization does not protect against advanced attacks, such as the Netflix movie recommendation dataset attack [27].

Privacy sensitive region:

Visual data may contain different privacy sensitive information. For instance, the face or the entire person may be regarded as privacy sensitive regions. Car license plates is another example of information that may be privacy sensitive. A privacy protection system will identify the privacy sensitive regions, for further anonymization.

Utility:

Utility of a dataset is the usefulness of the data for some given operation. Utility can be measured by how well the researcher can perform a particular task, given either the full dataset or an anonymized version of it [43]. This thesis defines face detection as one important utility. This utility can then be measured by performing face detection on both the original and the anonymized dataset.

2.1.2 Privacy protection methods

A privacy filter performs anonymization by removing information identity revealing information. A person can be identified by the face, but other visual clues like clothes, tattoos, and height can contribute to identification. People can even be identified by how they walk. This thesis will however focus on identification by and anonymization of the face. The face is the privacy sensitive region that will be anonymized. The most common group of privacy protection methods in image and video is called redaction. These are methods that modify sensitive regions. Padilla-López et al. have identified five categories of redaction methods, based on how the image is modified. Those are:

1. Image filtering.
2. Face de-identification.
3. Object replacement.
4. Object removal.
5. Image encryption.

Image filtering:

Filtering is based on image processing filters that modify privacy-sensitive regions. Image blurring, or smoothing, is a commonly used filter which is based on neighborhood averaging. The filter size and weights will influence the effect of the blurring. These filters can improve images by removing noise. But used as privacy filters, they must have a stronger effect to remove enough visual information. An averaging filter of size 9 × 9 is shown in 2.1. The filter is convolved over the image, and the value of the center pixel is replaced by the average of all the pixels under the kernel. In this example, all filter elements have the same weight to give the average of all 9 pixels. Both the size and weights will affect the effect of the filter.

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}. \tag{2.1}$$

Pixelation reduces information by lowering the effective number of pixels. An image is divided into a grid of square blocks. The average color of each block is calculated from all pixels belonging to that block. Both blurring and pixelation are frequently used in both TV and newspapers as privacy filters. Google Street View blurs peoples faces and other sensitive regions like license plates. Popular image processing libraries like OpenCV and Pillow have a rich set of methods for image filtering. The problem with both blurring and pixelation arises when privacy must be combined with intelligibility. Privacy preservation and intelligibility are two competing qualities, and compromises must be made. Research by Neustaedter et al. [28] studied always-on video links between distance separated co-workers. Particularly the case of office and home, where the home-office typically is a spare bedroom. This increases the risk for privacy-sensitive situations, as people can enter the room unaware of the camera. The main benefit of the video is awareness. Co-workers can see each other and evaluate if the other is available for conversation. They consider awareness and privacy as competing measures that must be balanced. Their study concludes however that blur filtering is not able to balance privacy and awareness for home situations. This should raise concern also about other video obfuscation techniques.

Face de-identification:

Face de-identification applies alterations to faces to conceal identity. The goal is to change a face in such a way that it will not be identified by face recognition software. This method has been used to combine privacy protection with the preservation of visual information. Before deep learning neural networks, the available methods were limited. K-Same family of de-identification algorithms were first introduced in the year 2005 by Newton [29]. A set of faces (face-set) is divided into clusters of size k images. The algorithm computes an average face based on the k images in the cluster. Then all faces in the cluster are replaced with the calculated average face. The algorithm provides *k-anonymity*, as the probability of recognizing a de-identified face is no more than $\frac{1}{k}$. The utility however was not guaranteed. Gross et al. [14] extends the K-Same algorithm to preserve utility like facial expressions and gender in face images. It does so by splitting a face image set into mutually exclusive subsets. K-Same algorithm is then performed on the subset, for instance male faces only. A problem is that the averaging takes place in pixel space, and misalignment's of the faces will lead to undesirable artifacts or blurry images. A different approach is to replace faces with photographs. In the year 2008, researchers in [2] built a large database of face images of different appearance and pose. A face is de-identified by swapping it with a similar face from the database. Face swapping offers high privacy protection and combined with high face detection rate. There are challenges however, as we want to maintain direction of head with a consistent looking face throughout the video.

Object replacement:

Object replacement method is based on replacing the privacy-sensitive region with a visual abstraction. Silhouette, polygonial model, and avatar are common visual abstractions. But also image filtering techniques and face de-identification can be classified as visual abstractions, as well as face swap.

Image encryption:

Image encryption is a method which protects the entire video or regions of interest. The correct key is needed to decrypt the video for view. The positive is that once decrypted, the video will appear in its original state with no loss. In the context of this thesis, a third-party machine-learning contractor would need the key to unlock the video for the machine learning training. The problem is that with the correct key, the video material is no longer anonymized, therefore this option is not further investigated.

Object removal:

Object removal techniques are used to remove the sensitive region with for instance inpainting. The image appears as if the object, for instance a person, did not exist. This method is found not suitable for protecting machine learning training data, as the very object to act upon is removed.

2.1.3 Evaluation of privacy techniques

The evaluation of the privacy protection system follows along two interdependent dimensions [9]:

1. The privacy protection level.
2. The utility of the technique.

Some methods are very strong privacy filters, like masking. But by removing much information, the intelligibility left in the image is lowered. But if this does not reduce the usability of the video, the utility is still preserved.

The effects and side effects of privacy techniques are explored in [1], and the authors proposed five crucial criteria for the optimum balance of privacy protection and the information loss in the privacy filter. Those are:

1. Efficacy: the ability to effectively obscure the privacy-sensitive elements.
2. Consistency: the short-term visual appearance of the subject is required to support object tracking. A moving object must appear consistent in such a way that this is possible.
3. Distinguishable: A privacy filter should not alter the subject to the point that the subject could not be distinguishable from other subjects within the same object class.
4. Intelligibility: Only protect the privacy-sensitive attributes and retain all other features/information.
5. Aesthetics: The privacy filter needs to maintain the perceived quality of the visual effects of the video-frame.

The consistency constraint requires that one individual in a video must be anonymized in the same manner throughout a scene in video. By the distinguishable constraint, every individual face must preserve some uniqueness. Individuals must be separable.

2.1.4 Object detection

This section explains the concepts of object detection using deep learning algorithms. Further, face verification and -recognition is explained. These are important subjects for this thesis. A privacy filter is depending on locating the privacy-sensitive regions to which the filter algorithm should be applied. The result can be evaluated using face verification and face detection. Some concrete implementations are also introduced. The recent image object detectors are largely based on deep convolutional neural networks.

An important distinction is made between *Image Classification* and *Object Detection*. The famous MNIST dataset is a typical classification task. This dataset has one handwritten digit per image. A classifier shall predict which class from a limited set of classes is correct. In this case the numbers 0-9.

Object detection is not constrained to one object per image. There can be multiple objects in an image and even different classes of objects. An object detector also needs to locate each object, usually by rectangle coordinates, called a bounding box. The detector may also return a class probability value. An important quality of an object detector is its accuracy in predictions. For this the actual value, or ground truth is required. The predicted value is compared to the ground truth. The ground truth consists of both location and the class type, for example car, person, cat, etc. The accuracy is calculated by comparing predictions to the ground truth. Accuracy and other related metrics are explained in 2.1.5.

Neural Networks

Neural networks are not a new invention, it's history dates back to 1943 when McCulloch and Pitts described how the basic processing elements of the brain worked, and demonstrated it using electrical circuits [5].

Neural Networks (NN) in the machine learning domain refers to *Artificial Neural Networks*, as opposed to biological neural networks of the brain. NN have evolved to become a major building block for machine learning algorithms. What is remarkably with NN is their ability to learn from data. An example that is hard or even impossible to program by hand, is to recognise handwritten numbers in images. By giving many examples of differently written numbers, together with their true value, the network can learn a function that maps an image to a number. When exposing a trained NN to new examples, the network is capable of predicting correct number with high precision.

The basic building block of a neural network is the single node perceptron. A perceptron takes an input X of size n, and has one output y. A weight w is associated with each input, and the there is a bias associated with the perceptron. The output is [22]:

$$y = f(\sum_{i=1}^{i=n} w_i x_i + bias). \tag{2.2}$$

An activation function f limits the output to the range $(0, 1)$. A much used activation function is the *sigmoid* function.

Training a neural network requires a loss function that measures the difference between the predicted result and the actual result, or target. The weights of the network are adjusted using backpropagation, and slowly the difference between predicted and actual results decreases.

The architecture of a NN can be tailored to specific problems. The number of inputs should match the number of variables in the specific problem domain. For instance, in the Iris flower dataset, there are four features: the length and width of the sepals and the petals. This dictates that there will be four input nodes. There are three types of the Iris flower, and the network should predict one of each for each sample data. This results in three output nodes. In each output node the networks predicted probability will indicate the type. A Softmax output layer is often used when the network shall predict only one of many. The Softmax layer sets the output with the highest probability to 1, and the rest to 0. A networks inner architecture can also vary to great length. With more complex data, higher accuracy is obtained by increasing the networks depth. This is done by stacking multiple layers of neural networks, forming a *deep neural network*.

An image of gray scale has two dimensions, (x, y). The one dimensional input of the traditional NN requires the image to be reshaped into a one dimensional vector. A problem with this is that neighboring pixels are split apart, losing spatial information. When analyzing an image, the neighboring pixels of both axes are important. *Convolutional Neural Networks*, also called both ConvNets and CNN, are especially suitable for images. It was first presented in a paper by LeCun et al. in 1998 [21]. It has since become the dominant architecture for image classification problems. A convolutional network is built to support the two dimensions of an image. In addition, color information is handled by adding depth, one CNN for each color. The CNN architecture is inspired by the visual cortex of the brain [33]. A CNN consists of a convolutional layer, an activation function such as RELU followed by a pooling layer. Usually several such elements are stacked after each other, creating a deep model. The network may be finalized by a fully connected layer, and depending on its usage, classification or regression, an appropriate last element is added. An overview of the image classification pipeline is illustrated in 2.1, illustration from the original paper by Yann LeCun et al.

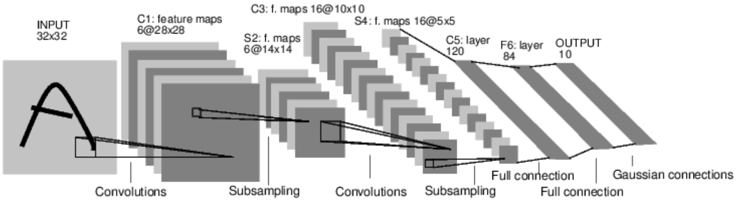


Figure 2.1: CNN image classification pipeline.

A rectangular filter, also called a kernel, is convolved over the image. This operation takes the dot product of the weights and the pixel values under the filter and produces one scalar value. The filter is moved over the width and height of the image. The result is called a feature map. Moving the filter one pixel at a time, stride=1, creates most overlap and the biggest output volume. On the other hand, increasing the stride gives other benefits such as fewer computations and faster performance. The balance must be tested out experimentally. One filter creates a two-dimensional feature map. It is common to use several filters, creating a depth the size of filters used.

A non-linear activation function is applied to the result of the convolve operation. Rectified Linear Units $RELU$ is a popular choice as it makes training several times faster than using *tanh* [19]. After a convolutional layer it is common to use a pooling layer. The pooling layer is used for downsampling, which is to reduce the spatial resolution. This in turn decreases the number of parameters the network has to optimize. Max pooling is the most common way of pooling, and it simply means that the highest pixel value is chosen. If the pooling filter is 3X3, then 9 pixels are reduced to 1. The downsampled output from one layer is the input to the next layer. In this way, feature maps are created that focuses on different things, from small to big structures.

Inception Network Accuracy can be improved by stacking more convolutional layers after each other. But there is a point where adding more layers does not help. Very deep networks are computationally expensive. They are also prone to overfitting. Another problem is that different object sizes in the image require different kernel sizes. Instead of just making the network deeper, Inception network [40] makes the network wider. Filters of different sizes operate at the same level. The inception network is built from several layers of inception modules. This network is also known as *googLeNet*. Later versions have further optimized the network For instance, it was found that two $3x3$ convolutions are faster than one $5x5$.

Residual Network Increasing the network depth increases the accuracy of the network, but only up to a point. Introducing residual blocks [16] made extremely deep networks possible. A residual block has a shortcut connection. This does not add to the complexity of the network. Stacking many residual blocks forms a *ResNet*. It was demonstrated that extremely deep resnets are easy to optimize, while their regular counterparts exhibit higher training error when depth increases.

Object detection plays an important role in an anonymization system. It is used to identity the privacy sensitive region, which should be anonymized. It is therefore essential that the detector has high recall. In this thesis a state of the art specialized face detector is used. But it is also possible to train a generic object detector using a labeled dataset. It is possible to benefit from a previously trained network. For instance, pre-trained weights for an object detector like YOLO can be used when starting to train on a different dataset. This is called transfer learning. Parts of the network can also be locked, so that only the last part actually is trained.

YOLO One of the leading object detection algorithms today is called YOLO [34]. YOLO runs on a neural network framework called Darknet. Using a GPU, videos can be processed in real-time. Its name You Look Only Once refers to that it performs only one forward pass through the network in detection mode. It is a multi-class object detector. This means it not only detect multiple objects in an image, but also predicts its class. For every detection, a probability score is calculated for each class. In version 1 and 2 of YOLO, the detection is classified as the class having the highest score. The latest revision 3 [35] as a more complex model for increased accuracy. But it comes with the cost of slightly lower speed. Another weak point of YOLO is its ability to detect small objects which are close together. This is also improved in version 3. The network works on three different scales, where the image is down-sampled by 32, 16 and 8. Another interesting change to version 3 is that detection is not restricted to being only one class. By removing the Softmax layer, detections become multi-class. Each class gets a score, and if above a threshold the object is considered to belong to that class. This is applicable in many settings, for instance detecting a person: A person can also be classified as male, female, young, adult, etc. This opens for many interesting detection applications using YOLOv3. Using YOLOv3 as a privacy filter it is possible to use different anonymization techniques

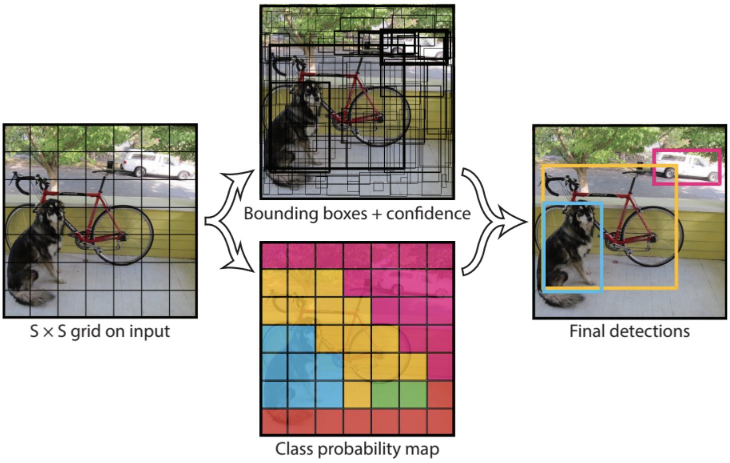


Figure 2.2: YOLO object classification.

for different object classes. For instance a face may be obscured differently than a car license plate. In this thesis YOLOv3 is trained for face detection using the WIDER-Faces dataset. YOLOv3 is 3.8× faster than RetinaNet at similar performance. But higher accuracy is achievable with RetinaNet. Another object detector alternative is Faster-RCNN [36].

Face recognition vs face verification

A face detector can not say anything about the identity of a detected person. To identify a person, there are two different approaches: Recognition and verification.

Verification is used when the question is: Is this the same person? Given image and id as input, determine whether the image is of the claimed person. Recognition tries to answer the question: Who is this person? An example, say there is a face database of K persons. Based on a given input image, output the identity of the image. Not recognized is also a valid output.

A possible solution to this task could be to train a CNN, and predict the identity through a Softmax output layer. There are at least two problems with this approach. Adding a new person requires changes to the network, and a following re-training. This would also require many images of each identity to achieve sufficient accuracy. A better option is to learn a similarity function $d(x1, x2)$. The similarity function maps a face image into a feature vector, also called embedding. The similarity function will cluster similar-looking faces nearby. Faces that are not similar will be further apart. The similarity between the two faces can now be measured by the distance between their feature vectors.

Distance: Norm of the difference of the embeddings of the images.

$$d(x_1, x_2) = \|f(x_1) - f(x_2)\|_2^2 \, . \tag{2.3}$$

This thesis uses a face detector called *MTCNN*. The Multi-Task Cascaded Convolutional Neural Network [45] can detect multiple faces in an image. It can also produce five landmarks per detection, giving the position of eyes, nose, and mouth. MTCNN consists of three convolutional layers: A proposal network (P-NET), a refine network (R-NET), and last is called O-NET. The P-NET is a shallow CNN that quickly produces candidate's windows. The second R-NET rejects a large number of false candidates. Finally, the O-NET further refines the results and produces five facial landmark positions.

The MTCNN is trained on the following three tasks

1. Face classification. A two-class classification problem: Face or not face.
2. Bounding box regression: For each candidate window, find offset from the nearest ground truth. Bounding box coordinates consist of left, top, width, and height.
3. Landmark position localization: The network outputs landmark positions. Training data has ground truth data, and during training the Euclidean loss is minimized.

There are 5 landmark positions: Left and right mouth corner, left and right eye, and the nose. The landmark positions can be used for alignment. To align face images means to position all faces so that the mouth and eyes are located at approximately the same position in all images.

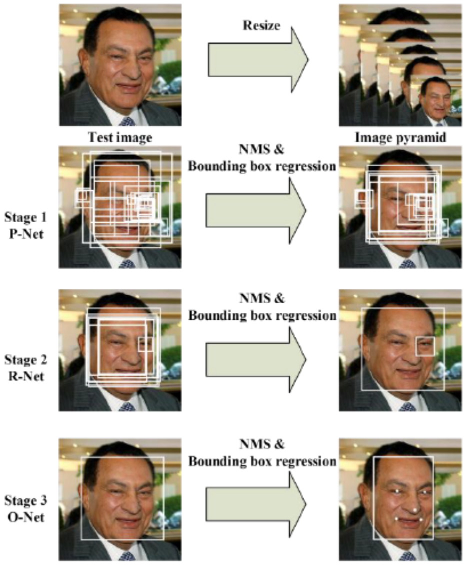


Figure 2.3: MTCNN pipeline.

Training a normal ConvNet requires many images per class. For learning face verification, it is more convenient to ba able to learn from few examples. Such system is called one-shot learning. An example of a one-shot learning system is the Siamese network [3]. Originally the Siamese network system was proposed for handwritten signature verification. Two identical neural networks are joined at the output. The layer before the last output layer of a neural network contains the "features", also called embedding. The embedding is a compact representation of the data the network has learned. Their Euclidian distance corresponds to their similarity. The verification process measures the distance between two embeddings. This method was later adapted for face verification in DeepFace [41].

A threshold τ separates the distance of the same person and different persons

Algorithm 1: Face verification using embedding distance.

```
if $d(x1, x2) > \tau$ then
    different persons ;
else
    Same person ;
```

FaceNet [38] introduced the triplet Loss. A triplet consists of an anchor, a positive and a negative. The anchor and the positive are images of the same person, while the negative is from a different person. The triplet loss function minimizes the distance between the anchor and the positive and maximizes anchor and negative distance. If randomly selected, many triplets would have negatives very different from the anchor. This would result in poor learning as the loss function would easily be satisfied. Instead, hard triplets are selected so that the network needs to separate similar-looking positives and negatives.

2.1.5 Object detection accuracy

A common way to evaluate an object detection algorithm is to compare the detected area with the ground truth area. The ratio between the intersection and the union of the two areas is called the Intersection over Union, IoU.

$$IoU = \frac{intersectarea}{unionarea}. \tag{2.4}$$

If the detected area overlaps the ground truth by more than a certain threshold, often 50%, the detection may be defined as a true positive, TP. The detection must also match the class of the ground truth, and its confidence score must be above a threshold. If all those requirements are satisfied, it is a true positive. A violation of the latter two makes it a false positive, glsFP. False negatives are objects that were not detected. Based on these numbers, two commonly used metrics can be calculated:

$$Recall = \frac{TP}{Totalpositives}. \tag{2.5}$$

$$Precision = \frac{TP}{TP + FP}. \tag{2.6}$$

2.1.6 Generative Models

Image generation is a field where there has been a rapid development in recent years. The latest systems can produce convincing images in high resolution. There are different classes of algorithms for learning deep generative models. In Autoencoder (AE), we find one of these fundamental architectures. It consists of an encoder and a decoder. It is quite common to use Autoencoder for unsupervised learning It aims at learning the features of the data. The encoder maps the input x into the latent space z. This is also called the bottleneck, as it is of a smaller dimension than the input dimension. The decoder is a mirror of the encoder and tries to reconstruct the input data from z. An AE is trained to minimize the reconstruction error. An autoencoder can learn an effective compression. But it does not work well for image editing. Samples in the pixel space are not necessarily close in the latent space. Changes in latent space can have consequences to unrelated parts of the image. In 2013 the auto-encoding variational Bayes or variational autoencoder (VAE) method was proposed. It imposes a prior over the latent distribution, an assumption that it follows some distribution. The images generated from a VAE tend to be blurry, a consequence of distributing probability mass diffusely over the data space [8]. While AE learns the relationship between dataset sample and latent representation directly, generative adversarial networks (GAN) learns this indirectly. GAN is a deep learning framework proposed by Goodfellow et al. in 2014 [13]. It has since gained much attention and research and produced increasingly better results. Two models, called the generator and the discriminator, are trained. They improve by competing against each other. The role of the generator is to produce an image and present it to the discriminator. If the produced image is of such quality that the discriminator predicts it comes from the training set rather than the generator, the generator is rewarded. The job of the discriminator is separate generated images from training data. Only one of the models can be trained at a time, to avoid training on a moving target. If the discriminator cannot predict correctly with more than 50% accuracy, the system has converged, and training should stop. If training continues, the feedback from the discriminator is now meaningless, and therefore the result of the GAN may collapse.

Only the discriminator has access to the training data. Better images produced by the generator will reduce the loss, which again adjusts the weights through backpropagation in the right direction. In this way, the generator slowly improves and produces better images. The early attempts produced blurry images, but many improvements have emerged. Another exciting new class of generators is the image to image translation. These networks learn a mapping from an input image to output image, and also the loss function to train this mapping [18].

Facial attribute editing is based on editing an image, not in pixel space as traditional image processing does, but in the latent space. The goal of facial attribute editing is to edit one or more attributes of a face. Arbitrary Facial Attribute Editing, AttGAN [17] is such a system, focusing on being precise in editing what you want and leave other details intact. There are 13 editable attributes, like gender, age, pale, and beard. A face has certain attributes, say “black hair”, “no glasses”. By setting expected attributes to for example “blonde hair” and “glasses”, the task is to modify the face to have these expected attributes, and not the existing. The editing process uses an encoder-decoder architecture. The encoder encodes the image into a latent space using the trained model. The latent representation of the face is conditioned on the expected attributes. This modified representation is then decoded using the same model, to produce an image. Fader Networks [20] offers a similar editing system.

2.1.7 Baseline

In this section two common traditional privacy filters will be tested for face recognition performance. The region to anonymize is found by expanding the rectangle provided by the MTCNN face detector by a factor of 1.4. By slowly increasing the filter effect, one can see where the filters reach acceptable anonymization and take a note of the face detection accuracy. A simple utility function is to take the product of the face detection and anonymization graphs. MTCNN was again used to detect the anonymized faces, and a white rectangle indicates detection. A reverse image search was used for anonymization checks. The search engines *Yandex* and *Google* were used. An image is classified as identified either the persons name is found, or the person appears the collection of similar images. The test was performed on 4 different individuals from the CelebA dataset with a total of 98 images. The same images will be used to test anonymization using the new privacy filter.

The blur image filter used the Gaussian kernel. The radius was increased from 1 up to 10, see figure 2.4 for two examples. Pixelation is accomplished by first resizing down to low resolution and then resize up again, see figure 2.5. The number of horizontal pixels varies from 24 in the first to 6 in the last.

With blur radius=2, half the images are detected using reverse image search.

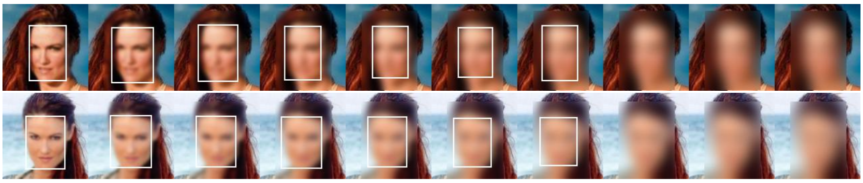


Figure 2.4: Blur image filter.

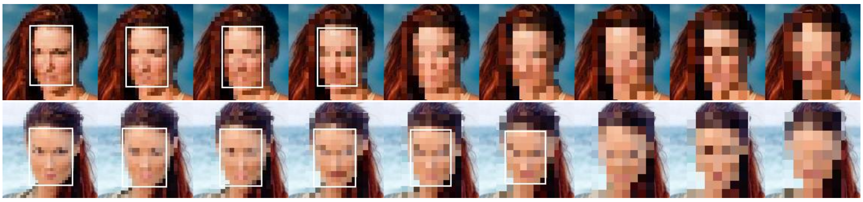


Figure 2.5: Pixelating image filter.

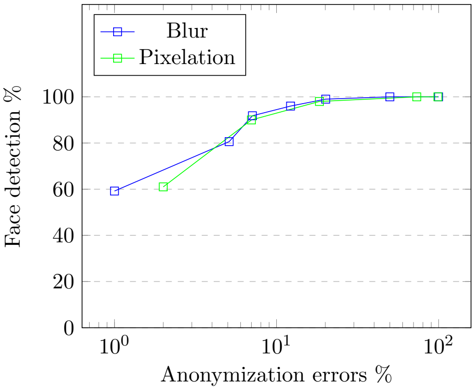


Figure 2.6: Anonymization errors vs face detections.

At radius=5, both utility and anonymization are at its highest. But the face detection rate has dropped to 92%. Two of the images were still identified at radius 8 and one even at the highest radius 10.

The x-axis in figure 2.6 shows anonymization errors in percentage, which is the rate of successful identifications. The two curves are remarkably similar. At 90% face detection, the anonymization error rate is 7%.

2.2 State-of-the-art

Researchers have proposed and applied various methods for anonymization by preserving the utility. So, in this section, we present current existing literature related to applying various methods and deep learning based generative methods on anonymization.

Flouty et al. [10] focuses on anonymization of video data, and how to achieve high recall. Video captured in the surgical operating room must be anonymized if stored on a server outside the hospital. They trained the Faster-RCNN to detect and blur faces. The WIDER dataset was used for training, but the faces in the operating room are very different, therefore a specialized labeled dataset was collected to refine the model. A recall below 90% meant many faces were not anonymized. To improve the recall they implemented a sliding average window which smooths out lost detections. This generates false positives, but the recall increased from 88% to 93.5%. A kernel size of 3 achieved the best performance.

Maniry et al. [25] proposed a privacy filter that obscures both the shape and appearance of privacy-related regions. The privacy-related regions are blurred according to three levels of privacy requirements. To improve intelligibility, the obscured region is overlaid with edges. The region is color-coded based on whether the activity is detected as normal (green) or anomalous (red). This gives high visual clues to interesting events through the use of color. The method was evaluated through a user study of eight different methods. This method achieved the highest privacy score, but the trade-off was intelligibility, which was below average. This is not only a privacy filter, but it is also an activity detector, using red colors to draw attention, and green for normal situations.

Midtun et al. [26] proposed an anonymizer targeted for persons involved in crime journalism stories. Only faces are anonymized, and the anonymized faces preserve a large degree of realism, while at the same time becoming more or less recognizable. The anonymization method works through morphing the source face with an average face. The anonymized images were evaluated by users according to two dimensions: face realism and degree of anonymization. While the images were found to look natural, respondents were able to recognize the persons in most cases.

Ren, Lee, and Ryoo [37] used an adversarial learning strategy to simultaneously optimize a privacy filter and an action detector. It consists of two competing systems: A video anonymizer that removes privacy sensitive information while trying to maximize action detection performance. The second system is a discriminator that tries to extract privacy sensitive information from the anonymized video. The method was tested for action detection and face recognition on data with actions involving the face, like brushing teeth, playing harmonica, applying makeup, and phoning. The new method was compared to several baseline methods: Blur, mask, noise, super-pixel, and edge. The new method achieved a better overall score on action protection and privacy protection. Blurring achieved the best overall score of the baseline methods.

UP-GAN by Hao et al. [15], was proposed as a utility preserving face anonymizer. The utilities it aims to preserve are age, gender, skin tone, pose, and expression. The model was trained using the UTKFace dataset [46], which contains 23708 images. From the dataset, they used 7 landmark points to detect pose and expression, and 4 attributes to preserve: age, gender, skin tone and pose. The UP-GAN generates a face based on landmark and attributes, and swap it with the original face. The model verified using a different dataset without landmark and attribute annotation. This was the FaceScrub [42]. Fake images were produced by detecting landmarks and setting a fixed set of attributes. The results showed that it outperforms traditional methods such as blur and pixelation.

The latest research using GAN based anonymization methods manage to achieve better results than the traditional image processing filters. The method by Ren et al. [37] which optimizes both anonymization and action detection is very interesting. Further work should be done to examine its feasibility as a generic anonymization method. This method would then anonymize the dataset before publishing it to external partners for further action detection training. UP-GAN is also a very interesting anonymization framework, where the pose and facial expressions are preserved. To do this, UP-GAN relies on 7 landmark positions. It is not clear from the paper what happens if the face is facing sideways, with only one visible eye.

Although various scholars have worked on anonymization, the anonymization method proposed in this thesis aims at preserving almost as much information as UP-GAN, except for age, skin color, and expressions. Whereas UP-GAN in [15] relies on 7 landmark positions to maintain pose and expression, our method needs landmarks for proper alignment. The MTCNN offers 5 landmark positions, and it is not clear if this is enough to provide proper alignment for all poses. To be able to preserve the gender attribute, both UP-GAN and our method rely on a gender detector. The biggest difference is the way the images are created. Our method uses a face editing network called AttGAN to apply changes to face, which should in theory preserve pose, and also enable consistency in visual appearance throughout a video. UP-GAN generates a face based on given landmarks and attributes. It is not clear if UP-GAN will provide consistent looking images as the pose and expression changes.

Chapter 3

Research Methodology

The chapter is divided into three sections. Section 3.1 describes the new privacy filter, while the evaluation methods is described in 3.2. The last section 3.3 describes improvement of the video object detector, Detection Short Term Memory.

3.1 Privacy filter

Then new anonymization method, or privacy filter, can be classified both as a de-identification filter and a face swap filter. An image, or video frame, is anonymized by first detecting all faces, then anonymize them individually. Finally, the modified faces replace the original faces. Privacy is achieved through changes applied to the face, and utility is maintained as the edited face is still natural-looking, and preserve pose and gender. Central to this solution is a face attribute editing framework called AttGAN. The AttGAN framework was created to edit one or more attributes while preserving the identity. But here it will be used to obscure identity and maintain a natural-looking face. By changing many attributes at the same time, the edited image is slightly changed overall. To preserve gender it is necessary to run the editing process two times. The first time gender is inverted, for instance from male to female. Age can be changed from young to old. By performing the edit in revers, gender is again inverted back to the original state, but the resulting face is not equal to the original. The second pass applies more changes, and if changed sufficiently, it can be said to be anonymized.

3.1.1 Dataset and data-preprocesing

The privacy filter will be tested using two different datasets. These are the publicly available CelebA [24] and VGGFaces2 [4]. These datasets contain face images harvested from the Internet, of more or less famous persons. The privacy filter will try to anonymize the persons. The CelebA dataset has in total 202599 images with 10177 different individuals. Each face is annotated with 40 binary attributes. The AttGAN is trained using 13 of the attributes of CelebA. The images are listed in random order. There are two versions of the images, a high-quality set and an aligned set. AttGAN was trained using the aligned dataset. The VGGFaces2 is less constrained than CelebA, having more variation in pose. Some images contain more than one face.

AttGAN expects an image size of 128×128 and normalized to the range $(-1, 1)$. The privacy-sensitive region, which in this thesis is the face, must be found using face detection. The AttGAN network expects as input the entire head including some margin. The face region from MTCNN is only a subset of the entire head. The region must be expanded, a factor of 1.4-1.5 was used. But if the person is facing sideways, the expanded area is offset, having more area in the front of the face, and not covering the back of the head. This indicates there is a need for a head detector, that includes the whole head rather than just the face. It is also necessary to align the face relative to its frame. The original alignment code for processing CelebA aligned images was not available. Instead, an alignment procedure was added to align a detected image similarly as CelebA. The alignment procedure use landmark positions from MTCNN, and centers the nose horizontally, and locates the nose at 65% from the top. A better alignment procedure would compensate for rotations also.

3.1.2 Implementation

The existing AttGAN [17] network will be used as the active editing component of the privacy filter. The editing process is as follows: Face image is cropped and adjusted for the AttGAN network, and enters the encoder. A set of pseudo-random attributes controls the editing process. The decoder transforms the latent space conditioned on the attributes to a new image having the desired attributes. The output is directed back to the encoder for a second pass. The attributes are reversed this time, that is, all values are multiplied by -1. The output from the decoder is the protected face image. The process is repeated for all face images in the image. The pseudocode is shown in 2.

Algorithm 2: Privacy filter pseudo code

```
Input: $x$
Output: $x_{anon}$
$detections, boxes \longleftarrow MTCNN.detect(x)$;
for $i = 0;\ i < boxes.length;\ i = i + 1$ do
    $gender \longleftarrow Genderdetector(detections[i])$;
    $attribs \longleftarrow Pseudorandom(gender)$;
    for $j = 0;\ \ j < 2;\ \ j = j + 1$ do
        $anon \longleftarrow AttGan(detection[i], attribs)$;
        $x \longleftarrow Paste(x, anon, box)$;
        $attribs \longleftarrow attribs * -1$;
    end
end
```

A copy of the image is made to become the anonymized image. The face detector locates all faces, and each face is edited one by one. Edited face is replaces original face. The final image has anonymized faces and only faces are modified. This algorithm shows a gender detector function, but this has not been implemented in the solution. A gender detector is rather assumed to exist.

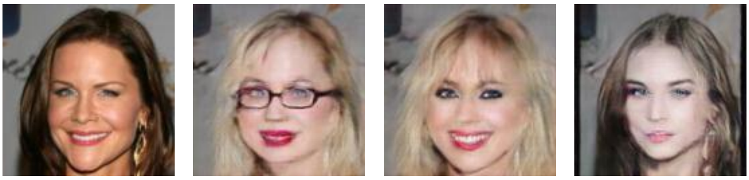
An input image x having n binary attributes a is fed into the generator to produce the latent representation z. See [17] for complete explanation of AttGAN.

$$z = G_{enc}(x^a). \tag{3.1}$$

b is the set of desired attributes. Editing is achieved by decoding z conditioned on b:

$$x^b = G_{dec}(z, b). \tag{3.2}$$

Equations 3.1 and 3.2 shows one forward pass through the encode-decoder



(a) Original (b) Edited1 (c) Edited2 (d) Edited3

Figure 3.1: Multiple anonymization results.

network. This process is repeated with the output of of 3.2 as input to 3.1, and $b = -b$.

3.1.3 Attribute Generator

In this thesis we will assume there exists a gender detector. This will enable us to synchronize several independent attributes that are gender-dependent. There are four of them: *Bald*, *Male*, *Mustache*, *No-beard*. Bald is set to a passive 0 during the experiments. The attribute generator creates a randomized set of combinations. This allows us to explore the possibilities of the editing function.

Figure 3.1 shows examples of editing using randomly selected attributes. Different attribute combinations create different visual appearances. By maintaining the same attributes for a particular face in video, the face will maintain the same appearance. The challenge is that we do not know which attribute combinations lead to good and bad results. By selecting randomly we will explore many different combinations. The random attribute generator produces a set of randomly chosen attributes based on gender, within a given range to limit the intensity of change. By exploring randomly, we can also expect attribute sets resulting in low anonymization, or poor visual quality.

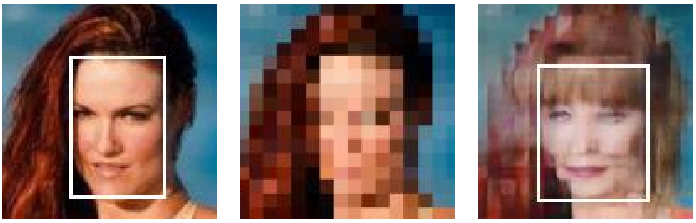
3.1.4 Privacy sensitive region

The aligned images of CelebA represents the ideal situation. The image contains one person, and the image is already aligned. This image can then be cropped on the center to 128 × 128 pixels. When the privacy sensitive region is found using face detection, the region is much smaller, see figure 3.2a. In the experiments, this rectangle is expanded to encompass the whole head. However, when the person is facing sideways, the detection rectangle only covers the front of the face, excluding the ears. When expanding the rectangle, much of the gained area is in the “air” in front of the person. An alignment procedure was added to center the face in the area of the expanded box. To do this, the landmarks from the MTCNN were used. The alignment procedure will maintain the position of the nose in the center in the horizontal axes, and at about 65% from the top in the vertical axis. The experiments show that persons facing sideways is a problem for the AttGAN network, possibly due to few examples in the dataset. This alignment does not compensate for tilted faces, that is when the head leans to one shoulder. This also causes poor results of the editing process.

3.1.5 Preprocessing filter

An experiment will be run to check if a pre-processing filter can lead to better anonymization while still maintaining utility. The idea is that a traditional blur or pixelate filter will reduce visual information into the AttGAN network, and thereby hide details from the output that otherwise might be preserved. An experiment will try to find the benefits and any drawbacks. Figure 3.2 shows process. The first image shows the face in full resolution. The next image is pixelated. This image is then anonymized by the AttGAN privacy filter. The result is shown in the third image. The white rectangles are the detection box of MTCNN, to illustrate that a pixelated face is detectable after a pass through the encoder-decoder network. When the image is pixelated as shown here, much information has been lost. The network is capable of reconstructing a face [7]. This is not the correct original face, and the result seems to be a good anonymization.

The experiments will increase the difficulty compared to the first test by using face detector to select the privacy sensitive region. The detected region must be expanded and the face must be aligned within the new cropping.



(a) Original (b) Pixelated (c) AttGAN edited

Figure 3.2: Preprocessing by pixelating.

3.1.6 Attribute learning

Using images from the less constrained dataset VGGFaces2 can reveal problems otherwise undetected using CelebA. CelebA is an "ideal" dataset, being aligned and having little variation in pose and head rotation. This experiment will use a face detector to locate the face, expand the rectangle to form the privacy-sensitive region to anonymize. Using the dataset's annotation file, the IoU can be calculated. The anonymization process should not affect IoU, and this will be measured.

The success of this privacy filter is to a large extent given by the chosen attributes. We explore attributes randomly to learn about the variation in visual quality and degree of anonymization. We do not know which combinations lead to good and poor results. By it might be possible to learn this, to control the attribute generator. To test this, a dataset will be created which links attributes to results. A machine learning algorithm will then be applied to learn to a mapping function from attributes to visual results.

Another interesting question is whether attributes leads to for instance bad result on all facial images, or if it is individual. This can be tested by visual pairwise inspection. If the classification of poor results does not match for the two different images, it indicates that attributes are individual.

We define a utility function to consist of distance, IoU, and face detection probability. We will test if this utility function can be maximized for a small group of individuals of the same gender. If it cannot be maximized for a small group, it means that it is individual. If this is the case, it means that maximizing the new privacy filter is more complex as it might require individual face analysis.

These experiments will try to answer research question 3 in 1.3.

3.2 Evaluation methods

This chapter will explain how we can link an objective measurement as distance to the question, is the face anonymized enough? The anonymized images will be evaluated on different dimensions. The goal of the privacy filter is to balance privacy and utility. Privacy will be measured in two different and independent ways 1) and 2). Utility will also be measured in two different ways 3) and 4).

1. Distance. Represents the change applied by the filter.
2. Reverse image search. Using search engine to verify anonymization.
3. Face detection rate. Protected faces shall be detectable.
4. Visual aesthetic inspection. Does the face look natural?

3.2.1 Privacy evaluation

The privacy dimension is measured using two different and independent methods: Distance and reverse image search. This will be used answer research question 2 in 1.3.

The amount of change applied by the filter is measured as the distance between the face embedding of the unprotected and the protected face image. The measurements are only applied to the detected region. The region outside of the detected face will not be used when calculating in the distance. This will ensure we are strictly comparing changes to the face. Including hair could increase distance more, and we could believe we achieved better anonymization than we actually do. A face verification system should not be fooled by a new hair style, or even a wig. The images in figure 3.3 indicates the region that will be analyzed for distance.

The second method is using a reverse image search of two search engine providers. These are Yandex.com and Google.com. Reverse image search is

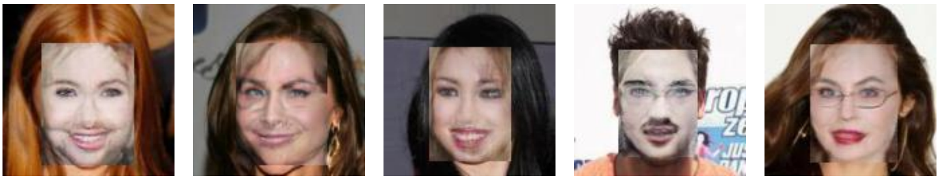


Figure 3.3: Distance measurement region

performed when the input to the search engine is an image. The classification fails if the search engine can find a match, either directly stating the person's name, or if the person is found in the collection of similar images. The two different search engines often identify different images. This method is time-consuming, and therefore limited testing has been performed. With two independent measurements, we can check how the distance correlates to the results of the reverse image search. The hope is to find a distance threshold, where anonymization can be regarded as sufficient.

All evaluation methods should be automated, both for saving time during experiments and to be able to compare the results of different studies. SphereFace-20 [23] was used in [37] to perform face recognition. Due to time constraints, this was not achieved in this thesis.

The similarity between faces are compared by measuring the distance between the embedding vectors. The embeddings are found using FaceNet with a model trained on the VGGFaces2 dataset. This method is pose and illumination invariant [38]. A test is arranged to find the following numbers:

1. Average distance between images of the same identity.
2. Average distance between images of two different individuals.

Four individuals were selected from the VGGFace2 dataset. There are 90 images of each individual. The table 3.1 shows the results. The first four rows are related to question 1 and the last three to question 2. These images are taken on different days, different poses, different lighting, etc. It is interesting to note that the average distance between random individuals of both genders are almost equal. When mixing genders the distance increase is minimal.

The leftmost plot in figure 3.4 is built from the distance matrix made from 100 different individuals, grouped by gender. Females are located in the

Folder	Count	Avg Distance	Description
n00452	90	0.653	Female
n00480	90	0.788	Female
n00527	90	0.626	Male
n00689	90	0.695	Male
Male	54	1.396	Mixed Male
Female	36	1.395	Mixed Female
Mixed	90	1.403	Mixed Male and female

Table 3.1: Distance distribution VGG faces.

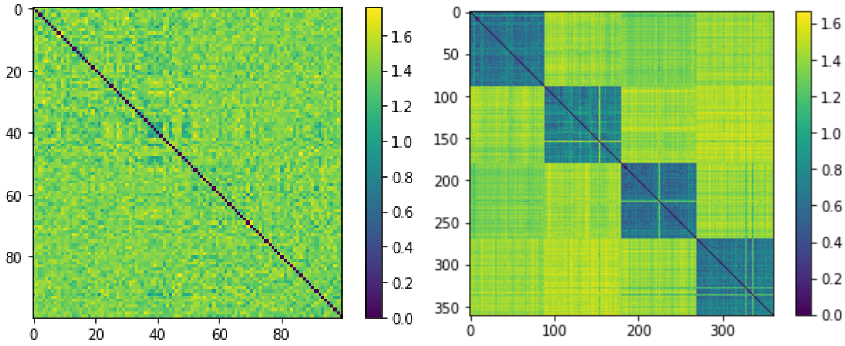


Figure 3.4: Distance matrix.

upper left quadrant and males in the lower right quadrant. The color-coding indicates the distance. The black diagonal shows a distance of 0, and happens when an image is compared with itself. The plot indicates some visually similar faces among the female group, but the average distance of both female and male faces are almost the same, at 1.358 versus 1.366. The figure on the right contains the four different individuals, 90 images of each. When an individual is compared with himself, the distance is less than when compared to other individuals. This effect is shown as dark squares along the diagonal.

3.2.2 Utility evaluation

There are several factors we consider as utility. Most importantly, the anonymized face should be detected using a standard face detector. This thesis use MTCNN. For face detection rate the recall is used. This is the fraction of predicted positives divided by total actual positives. An additional test will also check the accuracy of the detected region on edited images. This measures the change in IoU between original and edited image. For all experiments, a visual inspection is made. This will informally consider gender preservation, face direction preservation and aesthetic quality. This will be used answer research question 1 in 1.3.

3.3 Detection short term memory

Anonymization usually starts with locating the privacy sensitive regions in a video frame. Only detected regions are anonymized. This means that failing to detect leads to loss of anonymization. The identity of a person might be leaked from one unprotected video frame. The YOLO object detector processes videos frame by frame, with no knowledge preserved from the previous image. This thesis introduces a Short Term Detection Memory, STDM. This adds temporal memory to the object detector. A detected object will be preserved in memory from one video frame to the next. If the detector fails to detect the object in the next video frame, the detection will be restored from memory. The goal is to increase the recall of the object detector.

3.3.1 Functional description

The STDM operates on two lists of detections: The detection memory and the list of new detections which comes from analyzing the current video frame. For each video frame, all detected objects are added to the STDM. If a new detection overlaps with a detection in memory, the new detection replaces the old. When all new detections are added to STDM, the memory content is compared with the list of new detections. Detections in memory not found in new detections are added to the list of new detections. Each entry in STDM has a counter associated with it, which is increased when the detection is fetched from memory. A threshold value determines the number of consecutive times a detection can be added from memory. Once an object is detected, the counter is reset to 0. This will be referred to as the memory length. This was implemented in the Darknet object detector framework which YOLO runs on.

The Darknet framework does not have functionality for anonymization. Two different types were implemented: A blur filter and an object replacement filter. The blur filter uses Gaussian Blur function from OpenCV [30], and the object replacement filter inserts a selected image into the detected region.

To visualize the detection store in action, two changes are made to the Darknet framework: A detection that is restored from memory is highlighted with a red frame. A detection with low confidence, < 0.2, is highlighted with a yellow frame.

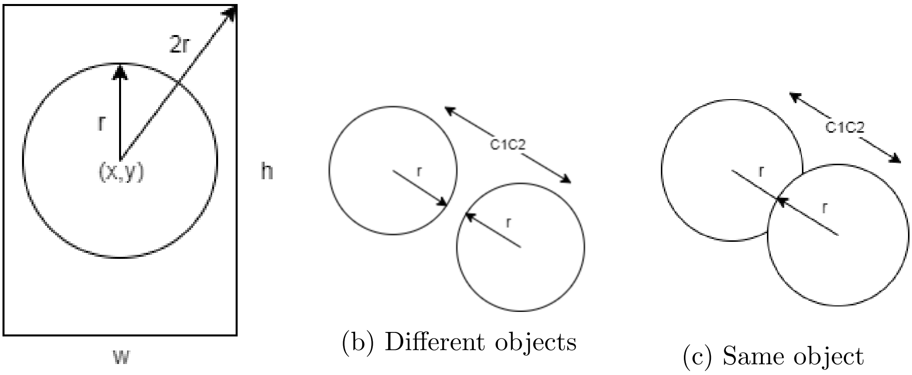
3.3.2 Datasets

YOLOv3 was trained to perform face detection using a subset of the WIDER Faces [44] dataset. YOLO documentation explains how to do it. The project site also contains pre-trained weights that should be used. The WIDER dataset contains many challenging images, with large groups of people. Even after training for about 14 days, the loss was about 10 times as high as preferred (5 instead of 0.5). But the goal was not to train a face detector, we only need it to experiment with.

The detection store must be tested on video. Four different videos were downloaded from [11]. Two climate change demonstrations in London, and two street camera videos from London and San Francisco.

3.3.3 Detection identification

Keeping track of detections introduces another problem: How do we compare detections? We need to know if a detection shall replace an existing detection in memory or if it is a new. A “collision detect” scheme will be used for this. Say there exist a detection d_1 in memory from the previous frame, and another detection d_2 is detected in current video frame. The new detection will always be added to memory. Now the system must determine whether detection d_1 is the same as detection d_2. If detection d_2 and d_1 does not collide, by comparing locations, they are defined as two different detections. d_1 is added to the current list of detections. But if the areas collide, they are defined to be the same object and we do not add. d_1 will instead be removed from memory. In this way, the memory is updated with the fresh detection. Instead of comparing the rectangles, a circle inside the



(a) Detection rectangle

(b) Different objects

(c) Same object

Figure 3.5: Collision detection.

rectangle was chosen for collision detection. This will make the system more willing to create new detections, to maximize anonymizations. But it may also increase the number of false positives.

The left image in 3.5 shows a rectangle with a circle inside. The rectangle represents the detection box with width w, height h and a center (x, y). The circle represents the collision detect area. If two circles intersect, or collide, the detection is defined to be the same. The new detection will replace the old. If the circles does not overlap, the objects are defined to be separate objects, see equation 3.4.

Collision detect algorithm: Two circles, radius r_1, r_2 inside detection boxes, having centers $(x_1, y_1), (x_2, y_2)$

$$C_1C_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{3.3}$$

$$C_1C_2 > r_1 + r_2 \tag{3.4}$$

Chapter 4

Experiments and results

This section describes the experiments that have been performed. The first three sections are dedicated to the privacy filter, and section 4.4 describes the object detector experiments.

4.1 Experiment1: Privacy filter

The methods used to evaluate privacy filter which are described in 3.2. Besides exploring the new privacy filter, we want to find the relation between the distance metric and the anonymization degree found by using reverse image search. The goal is to get confidence in the distance metric, in that way this number can determine when sufficient anonymization has been achieved. This experiment consists of three tests: First an initial test where we sample random attributes and evaluate the results. Then one image is anonymized 1000 different ways to measure the relation between distance and anonymization to more detail. Last, we apply a pixelation filter to the anonymization input pipeline to measure improvements, and any reduction in utility. The privacy filter is implemented with some additional monitoring capabilities, in order to collect data from the anonymization process. The attributes used are saved to file, and of course the anonymized image.

4.1.1 Anonymization and face detection

We measure detection recall, distance, and anonymization on images using random attribute sets. Distance and face detection recall was measured using a separate processing step after anonymization was completed. Anonymization is evaluated using reverse image search, and to make this more effective, only four different individuals are selected and can be seen in figure 4.1. These individuals are public persons and the search engines are very well equipped with images of them. In total 98 different images are selected. The individuals are:

1. 5941: Amy Dumas, wrestler
2. 6011: Lois C.K, comedian
3. 9739: Varun Dhawan, actor
4. 10154: Zhou Xun, actress

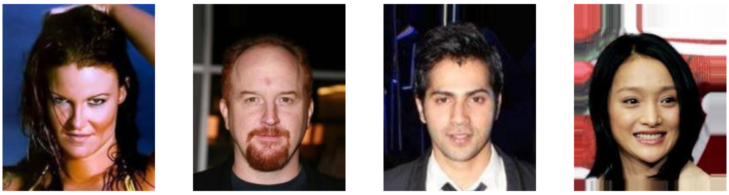
A batch is defined to be an image set of one individual, anonymized using a single attribute set. For each individual, 10 batches are run, creating a total of 980 images to evaluate. Each batch is initialized with different attributes from the random attribute generator. Also the distance between unprotected and protected images is calculated.

The setup for the experiment is considered as follows:

- Conditional randomized AttGAN symmetric attributes based on gender.
- The range is limited to $[-1, 1]$.
- No pre-processing image filter (pixelation).

The questions we seek to answer are:

1. What distance is produced? Is anonymization acceptable?
2. Face detection recall on anonymized face?
3. Consistency among images in batch?
4. Will gender be preserved in an aesthetically good way?



(a) 5941 (b) 6011 (c) 9739 (d) 10154

Figure 4.1: Faces used for anonymization.

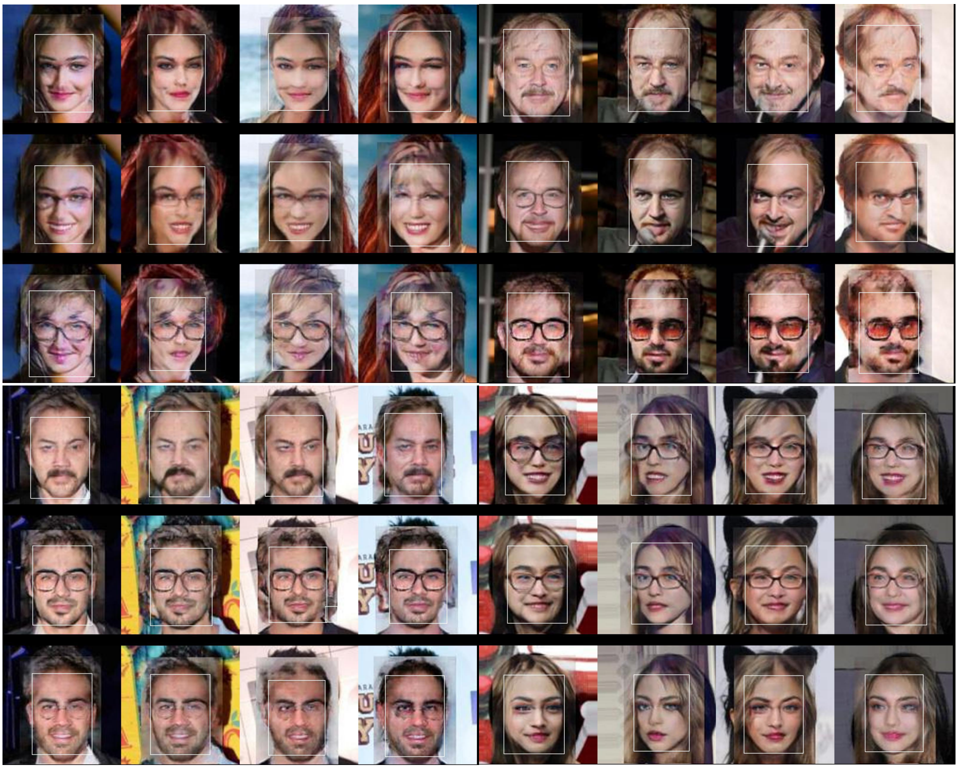


Figure 4.2: Anonymization using random attributes.

Figure 4.1 shows the four identities used in this experiment.

Figure 4.2 shows some samples. Images from the same batch are displayed in the same row. The white rectangles indicate the face detection rectangle.

The anonymization is performed consistently for the whole batch. This is important when anonymizing a video, the face must be consistent from frame to frame. Eyeglasses comes in different styles, and which one is added to a face is out of our control. It is a learned feature from the dataset. The same person can therefore appear with different styles of eyeglasses in the different images, even with the same attributes. In a video there will be less variation in the face from frame to frame, and will probably not be an issue. Gender is also preserved, but sometimes a mustache is visible on females. One noticeable drawback is that the shape of the head does not change much.

Anonymization

Evaluation of anonymization uses both the distance metric and the reverse image search as explained in 3.2. The Russian search engine Yandex identifies 100% while Google is lower at 80%. Also for the anonymized images, Yandex identification accuracy is better than Google. They identify different images, and in that sense they complement each other. A face is classified as identified if found by either of the search engines. The plot in figure 4.3 shows the anonymization success rate. Each batch consists of a unique set of randomized, gender preserving parameters for the AttGAN network.

Of the 980 anonymized images, the identity was detected by Yandex in 62 images. Two batches with identity 6011 accounted for 60% of the identifications. These two batches also have the lowest average distance.

For face id 6011, two batches have as low as 40% anonymization rate. This indicates that some combinations of AttGAN parameters are worse than others. But three batches have 100% anonymization. This indicates there is potential in the anonymization method. It is important to remember that the attributes used here were randomly selected.

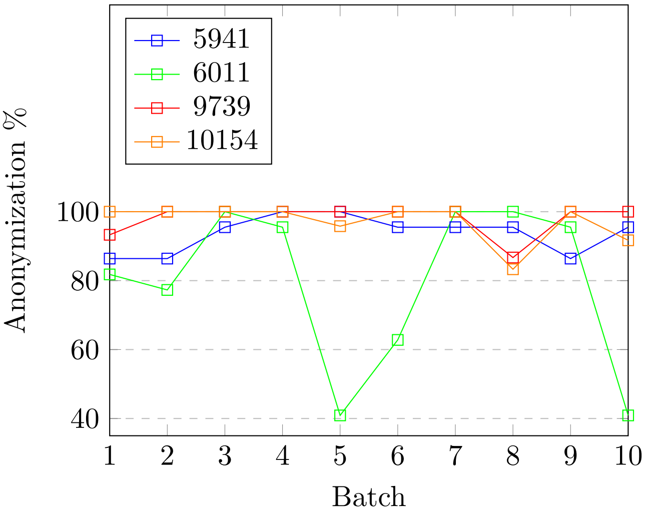


Figure 4.3: Anonymization success rate through the ten batches.

Distance

Face detection recall is 100% using MTCNN. This is an important factor for training data utility. The total distance average over every batch is 1.18%, and the maximum is 1.34. This is below the goal 1.4, 3.2.1. An interesting question now that we have distance numbers and their corresponding anonymization rate, is to check the correlation between them. Excel is used to make a table, with the average distance and anonymization rate for the 10 batches for each face identity, see figure 4.4. The two lows in the anonymization curve originates from the two bad batches for id 6011. The plots do show some correlation. Using Excel to calculate the correlation, using the function *CORREL*, we find the value to be 0.53. This is the same as the Pearson product-moment correlation coefficient. This will now be used to calculate the statistical significance. We claim that there is a relationship between distance and anonymization rate in this population, and the likelihood of being incorrect is set to 0.05. This is the alpha level. We use the table in [6] to find the minimum correlation coefficient. The degrees of freedom is $40 - 2 = 38$. The intersection of 0.05 and 35 is 0.325. If the correlation factor is above 0.325, we should accept the hypothesis. And at 0.53, it is.

Another way to look at the data is to plot the distribution. This is done by

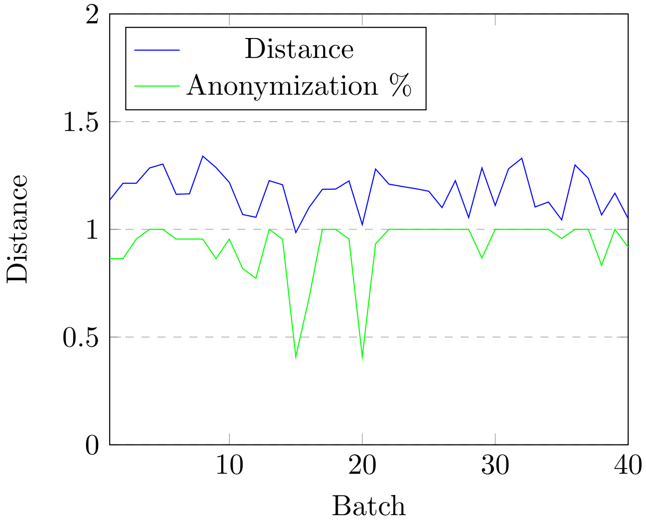


Figure 4.4: Distance and anonymization

range	[-3,-2)	[-2,-1)	[-1,+1)	[1,2)	[2,3)
Identified	5	17	20	1	0
Anonymized	1	16	125	33	2
Success rate	16%	48%	82%	97%	100%

Table 4.1: Distance distribution face id 6011.

calculating the mean and standard deviation of the distances for all the samples for this face identity. If data is normally distributed, the range within one standard deviation of the mean accounts for 68.3% of the samples. In the case of face id 6011, 65.9% (145 of 200) of the samples lies within this range. See table 4.1. The X-axis consists of the normal distribution ranges relative to the average. The center column is the range within one standard deviation σ of the average μ. Next column contains the range between $\mu + \sigma$ to $\mu + 2\sigma$ etc. The success rate increases as the distance is increasing. Knowing the anonymization success likelihood concerning distance can be important for a privacy filter. If a given parameter set gives too little distance, a new parameter set can be generated until a satisfactory result is achieved. Note that there is uncertainty in this measurement because of the low data volume. But it is intuitive that the less change from the original, the less anonymization is achieved.

It would be interesting to test the other male face using the same parameters. All parameters used were saved to a file so it is easy to replicate on a different image set. And it turns out that the anonymization result is poor also for face id 9739, with anonymization rate of 45% and 55%. By visual inspection it is also clear that the face has not changed much. So, from this, it can be concluded that the average distance produced from randomized attributes are too low. However, some batches show very good anonymization rates. A correlation between the measured distance and the reverse image search identification was confirmed. Preliminary tests show that attributes leading to low distance for one identity also lead to low distance on a different identity. However, more testing is required to confirm.

The anonymization efficacy from the previous test is not as high as desired. Instead of having symmetric parameters, generating a new set on the reverse transform can maybe create more variation. The reverse transform parameters still respects the gender parameter, just negated. The results show somewhat greater variation, and in some cases more extreme results. This occurs when one attribute pulls in the same direction for both transformations. The distance is not changed much on average, for two of the sets it was a little up and for the other set it was marginally lower. But the number of identifications were distinctly lower. In the first test, some images were identified up to six out of ten times. In this new test they were identified at most one out of times. At the same time 100% face detection on MTCNN is achieved. Successful anonymization is still dependent on selecting good attributes. Using this method still does not provide any guarantee for successful anonymization.

4.1.2 Distance vs anonymization

One image is anonymized in 1000 different ways allow us to measure anonymization rate based on distance in a more precise way. Mini batches are sampled from 5 ranges of the distance, each mini-batch consists of 30 consecutive images. Both Yandex and Google are used, and anonymization is classified as failed if the correct person is found by either of them. Both search engines finds the non-anonymized version of the image.

Table 4.2 shows the anonymization rate for a single image over different distance numbers. A major improvement on anonymization rate happened when distance is increased from 1.25 to 1.32. This table shows the potential

Avg Distance	Identifications	Anonymization rate
1.06	10	66.7
1.19	8	73.3
1.25	8	73.3
1.32	1	96.7
1.56	1	96.7

Table 4.2: Sampling anonymization in mini-batches of 30 grouped on distance.

of the anonymization method. By carefully selecting attributes, a distance of 1.5 can be achieved, which is above the target of 1.4 found in 3.2.

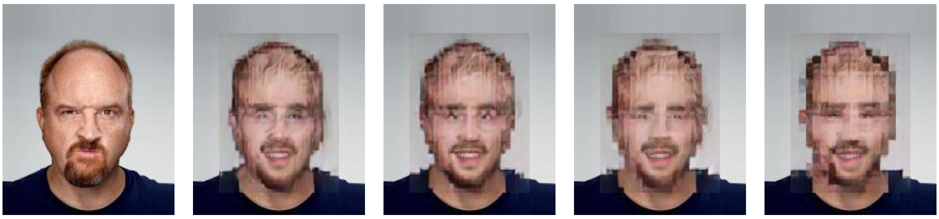
4.1.3 Pre-processing filter

The pre-processing filter is applied to reduce the visual detail level to the AttGAN network. The hypothesis is that the network can add details to the face, just as it can draw eyes when removing sunglasses. This experiment used face detection to locate the privacy-sensitive region. The face detection box was expanded by a factor of 1.4, defining the privacy-sensitive region to be anonymized.

Distance and face detection probability measurement is integrated into the privacy filter. The filter process saves attributes, distance, face detection probability to file. This experiment will use four different degrees of pixelation, identified by the superpixel count of the width. Ranked from low to high effect, these are 29, 25, 20, and 16. A random attribute set is created, and used for all four pixelation filters. This is done four times for each face identity, 5941 and 6011. There are 22 images for each identity, creating 176 images for each filter size.

Reverse image search was used to count identifications. To detect an increase in anonymization efficacy, the total number of identifications were used. For instance, one image is identified by both search engines for filter size 29, but only one for filter size 16. This indicates the anonymization has improved, although it is not perfect.

At the strongest pixelation, face detection recall had fallen to 80% for identity 5941 and 98% for identity 6011. The pixelation was performed in 4



(a) Original (b) 29 (c) 25 (d) 20 (e) 16

Figure 4.5: Pre-filter with pixelation.

Identity	29	16	Identification reduction
5941	26	15	42%
6011	38	25	34%

Table 4.3: Face identification count using pre-processing.

steps, however, the most interesting comparison is between the low (29) and high effect (16) filters. We are interested to know the effect the pixelation step has. Therefore, we account only for samples where the image is identified for pixelation size 29, and the face is detected for pixelation size 16. This will isolate the reduction in identifications to the pre-processing pixelation filter.

Table 4.3 shows the total identification counts for pixelation pre-processing filter 29 and 16. As all other factors are equal, the reduction in identifications can be linked to the pre-processing step. The gain in anonymization comes at a cost, as lost face detections and also lower aesthetical quality, see figure 4.5. It would therefore be better to first ensure the best possible attributes are used.

4.2 Experiment2: Accuracy

The metric Intersection over Union(IoU) was explained in the background 2.1.5. This experiment will measure IoU of detected face before and after anonymization. The change should be as little as possible. Ideally the predicted bounding box of the protected and unprotected image shall be the same.

Id	Batch	Number	TP	FP	Prec	IoU Change
n000001	1	254	240	10	96%	-12.5 %
n000001	2	254	250	8	96.9%	-8.5 %
n000001	3	254	250	14	94.7%	-8.5 %
n000009	4	90	84	1	98.9%	-5.6 %
n000009	5	90	80	4	95.2%	-16.9 %

Table 4.4: Accuracy measurements.

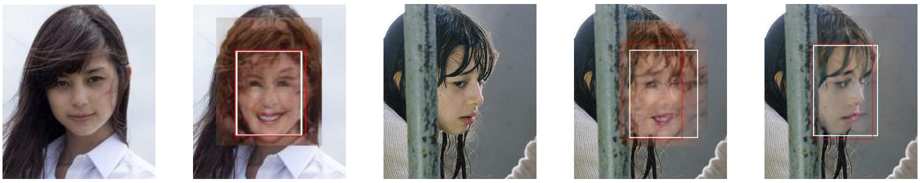
The dataset VGGFace2 fits the purpose of this experiment well. It is less constrained than the CelebA, in the sense that there is more variation in pose and rotation (tilt). However, some pre-selection of images are done to avoid problems on *IoU* measurements. The considered criteria are: Only one person per image, and the whole head must be within the image frame.

An extra alignment step was added to align the face after the first pass through the AttGAN network. This was done to improve sharpness, possibly the initial alignment was not good enough. The alignment process corrects the facial position relative to the bounding box, using the landmark positions from MTCNN. However, it does not perform rotational corrections, which would further improve the results. A rotational alignment correction can use the eyes landmark positions to rotate the face so eyes are in the same y-position. The whole anonymization process is monitored, and there are several possibilities for failure and those are:

1. Face not detected after first pass of AttGAN (used for 2nd. alignment).
2. Face not detected after second pass of AttGAN.
3. Multiple faces detected in anonymized image (false positive).
4. *IoU* less than 0.5 on anonymized face.

The attributes initially had a maximum range of $[-0.8, 0.8]$. Judged by the results, this is not sufficient for anonymization. For batch 5, the range was extended to the range $[-1.1, 1.1]$, except for the *Pale* attribute that was still restricted to $[-0.8, 0.8]$. This test shows that change in *IoU* varies with the attribute sets. This was not expected.

Table 4.4 shows the results of the experiment. In the table4.4 , Id refers to the image folder of the dataset VGGFaces2. An explanation of TP, FP



(a) Front orig (b) Front anon1 (c) Side orig (d) Side anon1 (e) Side anon2

Figure 4.6: Sideways head poor performance.

and *Precision* is given in 2.1.5. *IoU* change is the difference between the detected *IoU* on original and edited. This change is caused by the privacy filter. Some detections got *IoU* below 0.5, and where rejected.

It was surprising to find that the *IoU* changed so much. Most images have a relatively low change in *IoU*, but 18% have a difference of more than 10%. By identifying these images, it is noted that the problems originate in the unconstrained nature of the dataset. Tilted or sideways facing heads have a higher risk of poor results. For tilted heads, there is a rotational misalignment between the face image into the AttGAN network, and the facial images the network was trained on. This could be improved by rotational alignment on input, and the appropriate rotation back on output. For sideways facing faces, there is a tendency that the anonymization puts a forward-facing face, centered horizontally on the originals face nose. This results in two errors. The first one is the facial direction is not preserved, and the second error is that detected face is in the wrong position, and thus lowering the *IoU*. This is not consistent however, it is more frequent in some batches than others.

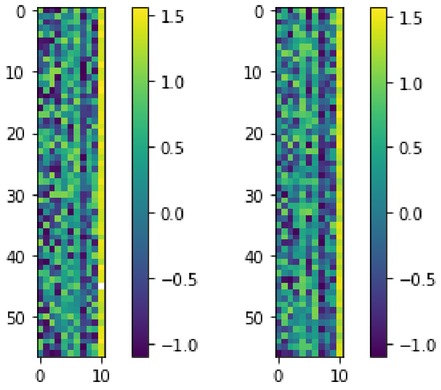
Figure 4.6 shows an example where editing a sideways directed face fails. The attributes used for the images *b* and *d* are the same. The image *e* is anonymized using a different attribute set. The red rectangle represents the ground truth, and the white rectangle the detection.

4.3 Experiment3: Learning attributes

Previous experiments show that some attribute sets are better than others. Poor attributes leads to two types of problems: Poor visual appearance and poor anonymization. Experiment 3 consists of several sub experiments,

Batch	Samples	Bad	Ok	Good
1	2000	212	1118	670

Table 4.5: Learning parameters.



(a) Bad (b) Good

Figure 4.7: Attributes vs distance plot.

tests, to investigate the relation between attributes and the anonymization outcome.

4.3.1 Predict visual appearance

This test focused poor visual appearance. The goal was to see if machine learning algorithms could predict the outcome of attributes. A single image was anonymized 2000 different ways, using the gender preserving random attribute generator. The range was set to $[-1.1, 1.1]$, except for the *Pale* attribute which was limited to $[-0.8, 0.8]$. Symmetric attributes simplifies the dataset, requiring only one attribute set per image. The experiment writes the attributes, IoU, and distance to file . Images are graded by visual inspection from bad (0), OK (1), to good (2), and the result added to the file. The grading is only for the visual appearance. This file will be used as dataset for learning the mapping between attributes and visual appearance.

The figure 4.7 shows a sample of attributes that lead to poor visual results

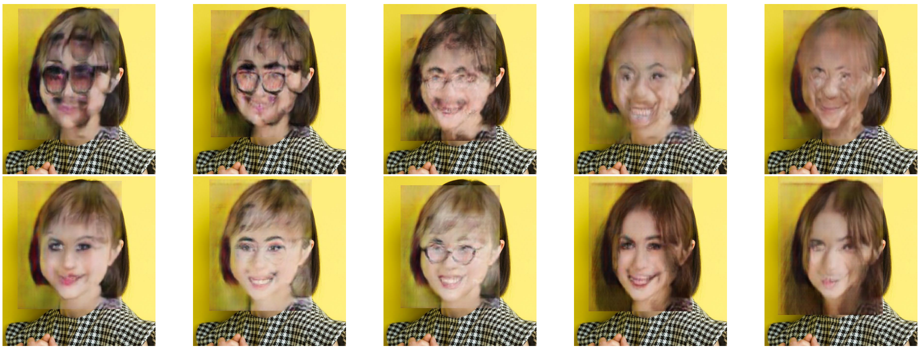


Figure 4.8: Classification examples of bad (top) and good (bottom).

which can be seen in figure 4.7 a) and to good results which can be seen in figure 4.7b). The plot shows the 10 attributes actively used, and the rightmost column is the distance.

Figure 4.8 shows a few examples of classifications. Distinctly good and bad results are easy to classify, but borderline cases could be classified differently. The classifications are subjective, and may not be the same if repeated. With the labeled dataset, this becomes a supervised learning problem. A deep neural network can learn non-linear models. Scikit-learn [32] has the MLPClassifier, a multi-layer perceptron configurable network that trains using backpropagation. The dataset was graded with 0,1,2. The target was set to 2 (good), making this a classification problem. Data is split into 70% for training and 30% for testing.

The input layer is given by the number of features, which is 10. The hidden layer architecture is not so obvious, and a good candidate is found through experimenting. The chosen network configuration achieved 77% accuracy with this network architecture, all fully connected layers i.e., (10,20,40,20,20).

Once the model is trained, the weights can be saved and used in the attribute generator. The model can predict the outcome of the randomly generated attributes. A new set of attributes is generated until predicted to be good. On average it takes 3.1 tries to get an accepted attribute set.

A new batch of 1000 edits using the same image was created to check if there were improvements. The interval for the pale attribute was increased from $[-0.8, 0.8]$ to $[-1.1, 1.1]$. This resulted in more white faces. There was distinctly less samples of visual poor quality, counting 29. These can roughly be grouped into two classes: Too white and hairy face. When applied to batch of processing of images, the improvement is not so much reflected in the numbers. The average change in IoU is not improved, probably because rotated and sideway faces still is a problem, the recall marginally better with 99.7% vs 99.2%. The number of internal failures has dropped from 0.8% to 0.3%. These are cases where the edited face is not detected within the privacy filter, for the internal alignment.

This training dataset was limited to only one face identity, and we do not know if this trained model will perform well on different faces. However, this experiment indicates that machine learning can help generate attributes leading to good results.

4.3.2 Testing the model on different faces

This tested whether attributes leading to bad results on one image also leads to bad results in others.

Two images of different persons A and B are edited 1000 times. Both images are subjected to equal attributes. Bad results for person A are identified and compared with the corresponding result for person B. In most cases the results agree, but 14% of the cases were classified as OK or even good. This indicates that attributes should be selected based on analysis of the face.

4.3.3 Maximizing a utility function

This test was set up to see if a utility function could be optimized for a group of images, all having the same gender. A dataset was created with the following individual parameters as the utility function:

1. Face detection probability p, interval $[0, 1]$
2. IoU, interval $[0, 1]$
3. Distance, interval $(0, 1.6)$

The utility function calculates the product of these parameters. Eight same-gender images of different individuals are selected from the CelebA dataset. One batch edits the eight images with equal attributes. For each image the following is written to dataset file: Attributes, the three utility parameters and their product labeled *Utility*. 1000 batches are run, producing 8000 dataset entries. To test which attributes leads to high utility numbers, we set a threshold on the utility parameter, the learning process is changed for classification instead of regression. Two numbers are selected i.e., "Ok" is above average, and "Good" is above 75%. The numbers are 1.04 and 1.15.

1773 samples were classified as "Good" and 4002 as "Ok". The result of this model is very poor when setting the target to "Good" and including all individuals. The true positives are very low. From the test set of 1600 samples, only 12 true positives are predicted, and 1221 true negatives. The model performs better when trained on individual images.

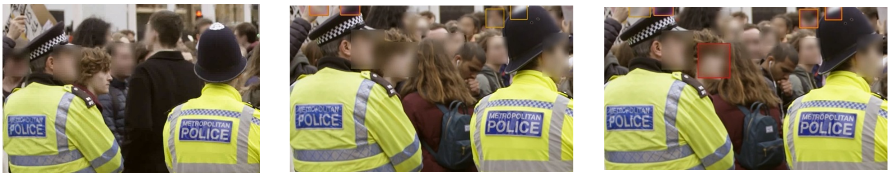
Image	Recall Ok	Accuracy Ok	Recall Good	Accuracy Good
1	78%	69%	66%	79%
2	91%	82%	74%	69%
3	62%	62%	40%	80%
4	57%	69%	27%	93%
5	42%	56%	20%	84%
6	65%	81%	0%	84%
7	48%	66%	25%	89%
8	63%	62%	39%	77%
All	59%	58%	3%	77%

Table 4.6: Maximizing utility function.

The table 4.6 shows the prediction accuracy for two different thresholds of the utility function.

It turned out that we cannot optimize utility for images of different individuals. The recall of only 3% shows this. The optimization works better on individual images, but there are big individual differences in achieved recall. When setting the target lower, above average, the model performs better. It is possible that settling for "good enough" can be an acceptable strategy. This may still prevent the bad results that can lead to loss of detection.

Experiment 3 has showed that each face need to analyzed to be able to find its optimum anonymization attributes. This has not been done. This makes the anonymization process more complex. It is also possible that the existing attributes of the CelebA dataset can be helpful for this. We also



(a) Undetected (b) Detected (c) Memory

Figure 4.9: Detection restored from memory.

mapped distance to anonymization in 4.1.2 in a more precise way, giving more confidence to the distance metric as a quantification of anonymization.

4.4 Experiment4: Detection Memory

The Short Term Detection Memory (STDM) is added to the YOLOv3 object detector. The change is implemented in the Darknet neural network framework, which YOLOv3 runs on. It is also extended with two anonymization methods i.e., blur and mask. YOLOv3 was trained for face detection on the WIDER-Faces dataset. Example videos were downloaded as described in 3.3.2. In this experiment, the videos will be used to test the effect of STDM.

Figure 4.9 shows details from the video. In person standing sideways in image 4.9a) is frequently undetected. In image 4.9b), he is detected and therefore also anonymized. The last image 4.9c), the red frame indicates that the detection was inserted from memory, and not from the detector. This means the face can still be anonymized, even if the object detector missed.

With this experiment, we will measure how recall and accuracy changes with different memory length. The results will depend on the scene, objects being static or moving. A detection that is fetched from memory will be inserted into the previously registered position. If the object moves, the precision will decrease. If the object moves fast, the detection inserted from memory will represent a false positive.

The experiment requires the ground truth to measure accuracy. The downloaded videos 3.3.2 are of course not annotated. This will be done as follows:

1. YOLO_Mark is used to convert video to images, see C.1.
2. Darknet command produce annotation files, by detecting faces in the images.
3. Manually edit annotations using YOLO_Mark, as not all faces are detected.

The effect of the STDM was measured by anonymizing a video using increasing memory length. As the object detector is not perfect, there will be missed detections. With increasing memory length, more faces will be anonymized. There will be fewer faces left to detect. By counting the number of none-anonymized faces, we can measure the effect of the detection memory. Faces are counted using the Darknet “test” method, which performs object detections in images. The anonymization was performed with object threshold parameter $-thresh = 0.24$ and the detection with $-thresh = 0.20$. This should enable detecting more images than were anonymized.

Method in sequential steps:

1. Create one image for each frame in the video using YOLO_Mark.
2. Create file with path to each image.
3. Run Darknet detection test command on the file generated in 2. Use option -save_labels to write detection results to files (one file per image)
4. Count number of detections per image. This is done by counting lines in files produced in 3.
5. Make another anonymized video with increased memory length. Repeat from point 1.

Figure 4.10 shows how faces have been anonymized using green masking. The thick red frames show faces detected by the object detector. This experiment counts the number of such detections using varying memory length. A strange behaviour of the object detector was discovered in this experiment. More faces were detected from images than from the video. Even though the threshold was lowered for the image object detection, the increase is larger than expected. Three videos with varying dynamic content, from

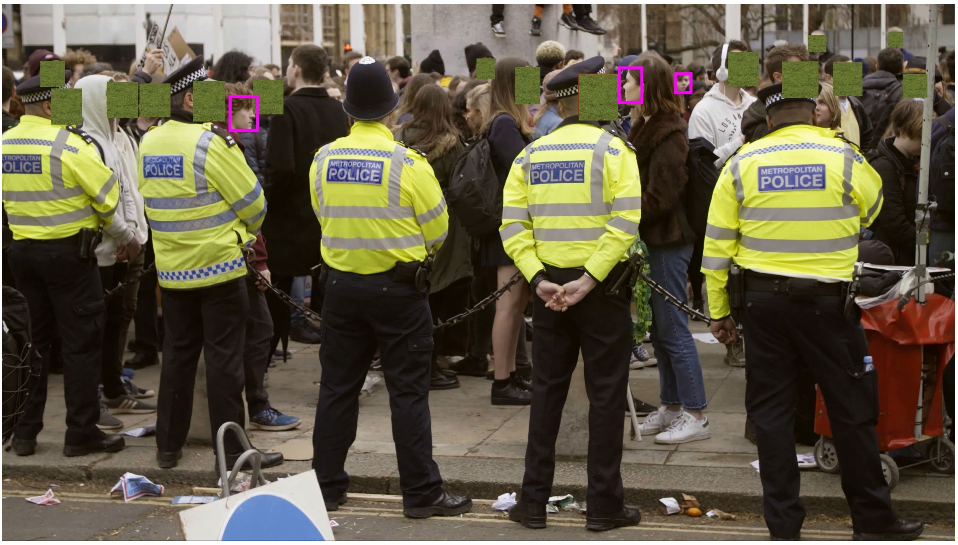


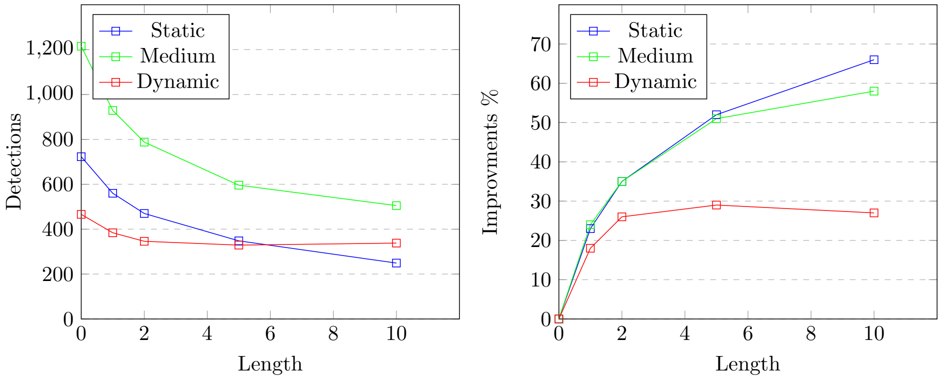
Figure 4.10: Face detection on anonymized video.

static, medium to dynamic. Dynamic content involves cars, buses and bicycles. Medium dynamic content is persons walking. This video represents the static content. People are standing still or moving slowly.

Figure 4.11b show the improvements as a function of memory length. Repeating the same test for other videos having more dynamic content. The curve is about the same, but the amount of false detections increases rapidly on dynamic videos with increasing memory length. This implementation of DS does not track movements, and thus a detection inserted from memory is stationary, while the actual object has moved. This will result in lower accuracy.

Testing on another movie clip with more dynamic content, London street camera. Fast moving objects like persons on bicycles can create false positives. The person has moved so fast that when comparing old and new detections, they do not overlap (no collision). Therefore the old detection is preserved in memory and creates a false positive. Without object tracking, the detection memory length must be short for fast moving objects.

Measuring accuracy and recall: The previous experiment did not take into account Intersection over Union (IoU) to measure accuracy. This requires the ground truth. Annotating the videos is time consuming, and only



(a) Detections.

(b) Improvements.

Figure 4.11: Measuring effect of STDM.

a limited amount of video data was annotated. Using movie with static content 4.10

In figure 4.12, the recall and mean average precision (mAP), are measured on the diminishing amount of detectable faces in the anonymized videos. As more faces are anonymized with increasing memory length, the object detector detects fewer faces. For this experiment therefore, the falling curve indicates better performance of the anonymization. The ground truth indicates the location of all faces, but the anonymized face are not found. Using the mask method effectively stops face detection from succeeding.

Based on an overall judgement, a memory length of 2 is the best. This gives a good increase in performance (both mAP and recall) and does not clutter the video with outdated detections. Objects that move will be anonymized less precise, as the detection store does not predict the objects new position. This can be improved by incorporating object tracking to the detection store, to enable the prediction of the next location. This result seems to be in line with the finding in [10], which suggests a short sliding window.

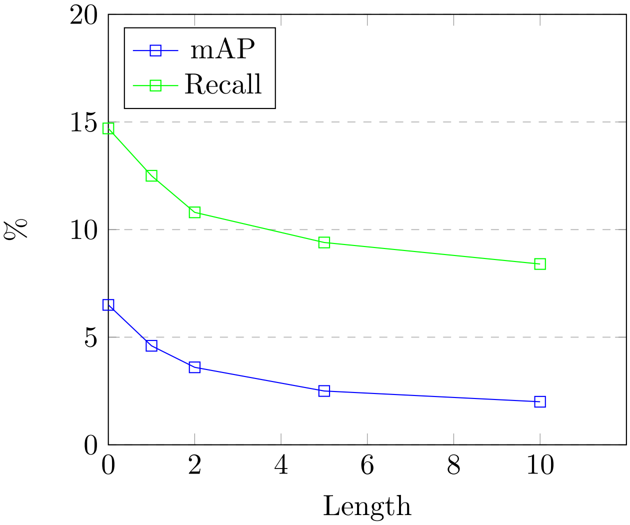


Figure 4.12: mAP and recall vs memory length.

Chapter 5

Conclusion and Future Work

This chapter summarizes the achieved goals and results in this thesis. It also describes identified future work.

5.1 Conclusion

In this thesis we propose an encoder-decoder based approach for anonymizing faces in visual media such as images and video, while at at the same time preserve the data's utility. Its intended use is to anonymize visual data for police-related action-detection tasks. To be useful as training data, anonymized faces must be as detectable as their non-anonymized counterparts, and we have aimed to preserve a human looking face, and preserving the perceived gender. A weakness with the experiments is that a face recognition was not implemented. This would have allowed for finding comparable numbers using the same dataset as other research. The following paragraphs concludes each thesis goal.

A face is anonymized using two forward passes through the AttGAN encoder-decoder network. The attributes controlling the editing process have been explored through a randomized process. The results show a great variation, both in terms of visual difference from the original, and also in visual aesthetics. The methods potential is shown when batches of images are anonymized to near 100% and still achieve 100% face detection.

An objective measure of anonymization is difficult. In this thesis, the distance between original and anonymized images has been compared to the anonymization rate using reverse image search. By applying a change equal to the average distance between randomly selected persons, high anonymization rate is achieved using reverse image search. Experiments shows that such distance, or amount of change is achievable, but it is a challenge find these attribute combinations. The potential of the system is demonstrated by achieving distance above 1.5.

The outcome is dependent on finding one attribute combination that leads to good results. An experiment showed that it is possible to predict the visual appearance of an anonymized face based on the attributes. However, experiments also show the optimum attributes can only be selected after an initial analysis of the face. Experiments reveal that the method performs poorly on faces having sideways pose or rotation/tilt. This requires several improvements: Precise detection of the head, AttGAN must be trained support editing sideway facing faces, and the need for an improved alignment which supports rotational corrections.

The introduction of short term detection memory increased the number of anonymized faces in video. The best results were achieved for relatively static objects.

5.2 Future Work

Two evaluation methods needs to be implemented for effective experimenting: A face recognition system that evaluates the anonymization, and evaluation of the visual aesthetics of the anonymized face.

More research is needed to predict and generate good parameters. The CelebA dataset has a great set of annotations, and it might be that attributes needs to be set according to existing attributes, as we already set gender dependent parameters.

The editing process now requires two passes through the network in order to apply enough change, and at the same time preserve gender. This should be reduced to only one pass. A possible solution to this is to train an anonymizer and action detector in an adversarial manner [37], and use AttGAN or similar to apply changes to faces.

Train AttGAN on an additional dataset. It is currently trained using CelebA. Even though it contains 200000 images, they are limited in pose (looking into camera), age (adults only), activity (celebrities posing) etc.

The introduction of short term detection memory improves the recall, best results achieved for relatively static objects. If the next location could be predicted, the accuracy would be higher. An action tracking framework should, therefore, be used in combination with a detection preserving memory. Also, a tracking mechanism is required to anonymize faces in video consistently.

References

[1] Badii Atta, Ahmed Al-Obaidi, Mathieu Einig, and Aurélien Ducournau. Holistic privacy impact assessment framework for video privacy filtering technologies. *Signal & Image Processing : An International Journal*, 4:13–32, 12 2013.

[2] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree Nayar. Face swapping: Automatically replacing faces in photographs. *ACM Trans. Graph.*, 27, 08 2008.

[3] Jane M. Bromley, Isabelle Guyon, Yann LeCun, Eduard Sackinger, and Roopak Shah. Signature verification using a siamese time delay neural network. In *7th Annual Neural Information Processing Systems Conference*, pages 737–744. Morgan Kaufmann Publishers, 1994. Advances in Neural Information Processing Systems 6 Edited by Jack D. Cowan, Gerald Tasauro, Joshua Alspector.

[4] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 67–74, 2018.

[5] Jack D. Cowan. Discussion:mcculloch-pitts and related neural nets from 1943 to 1989. *Bulletin of Mathematical Biology*, 52(1):73 – 97, 1990.

[6] Ph.D. Del Siegle. Critical values of the pearson product-moment correlation coefficient, 2015.

[7] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. *CoRR*, abs/1411.5928, 2014.

[8] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference, 2016.

[9] Ádám Erdélyi, Thomas Winkler, and Bernhard Rinner. Privacy protection vs. utility in visual data. *Multimedia Tools and Applications*, 77(2), Jan 2018.

[10] Evangello Flouty, Odysseas Zisimopoulos, and Danail Stoyanov. Faceoff: Anonymizing videos in the operating rooms. In Danail Stoyanov, Zeike Taylor, and Duygu et al. Sarikaya, editors, *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*. Springer International Publishing, 2018.

[11] hosted by Videvo.net Footage courtesy of Videvo.

[12] Gartner.com. Gartner predicts outdoor surveillance cameras will be largest market for 5g internet of things solutions over next three years, 2019.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[14] R. Gross, L. Sweeney, F. de la Torre, and S. Baker. Model-based face de-identification. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 161–161, 2006.

[15] Hanxiang Hao, David Güera, Amy R. Reibman, and Edward J. Delp. A utility-preserving gan for face obscuration. *ArXiv*, abs/1906.11979, 2019.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[17] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, PP:1–1, 05 2019.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[20] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. *CoRR*, abs/1706.00409, 2017.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[22] Mauro Castelli Leonardo Vanneschi. Perceptron.

[23] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition, 2017.

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[25] Dominique Maniry, Esra Acar, and Sahin Albayrak. Tub-irml at mediaeval 2014 visual privacy task: Privacy filtering through blurring and color remapping.

[26] Bjørnar Tessem Simen Karlsen Lars Nyre Midtun, Joar. Realistic face manipulation by morphing with average faces. *Norsk Informatikkonferanse 2017*, 2017.

[27] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006.

[28] Carman Neustaedter, Saul Greenberg, and Michael Boyle. Blur filtration fails to preserve privacy for home-based video conferencing. *ACM Trans. Comput.-Hum. Interact.*, 13:1–36, 03 2006.

[29] E. M. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):232–243, 2005.

[30] opencv.org. Opencv.

[31] José Ramón Padilla-López, Alexandros Andre Chaaraoui, and Francisco Flórez-Revuelta. Visual privacy protection methods: A survey. *Expert Systems with Applications*, 42(9):4177 – 4195, 2015.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. PMID: 28599112.

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015.

[35] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.

[37] Zhongzheng Ren, Yong Jae Lee, and Michael S. Ryoo. Learning to anonymize faces for privacy preserving action detection, 2018.

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[39] N. Sulman, T. Sanocki, D. Goldgof, and R. Kasturi. How effective is human video surveillance performance? In *2008 19th International Conference on Pattern Recognition*, pages 1–3, 2008.

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[41] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE*

Conference on Computer Vision and Pattern Recognition, pages 1701–1708, 2014.

[42] Vintage Vision & InterAction Group. Facescrub: A dataset with over 100,000 face images of 530 people.

[43] Felix T. Wu. Defining privacy and utility in data sets. *Norsk Informatikkonferanse 2017*, 2012.

[44] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[45] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks, 2016.

[46] Song Yang Zhang, Zhifei and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

Appendices

A Hardware Specification

Operating System	Windows 10
Processor	Intel i7-4900MQ
Memory	32GB
Graphics	1x NVIDIA Quadro K4100M

Table 1: Hardware specification

B Anonymization

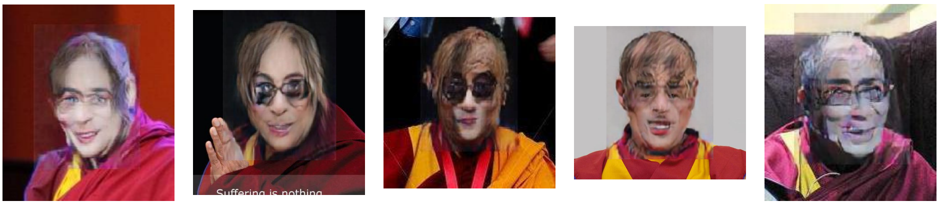


Figure 1: VGGFaces2 examples: Dalai Lama

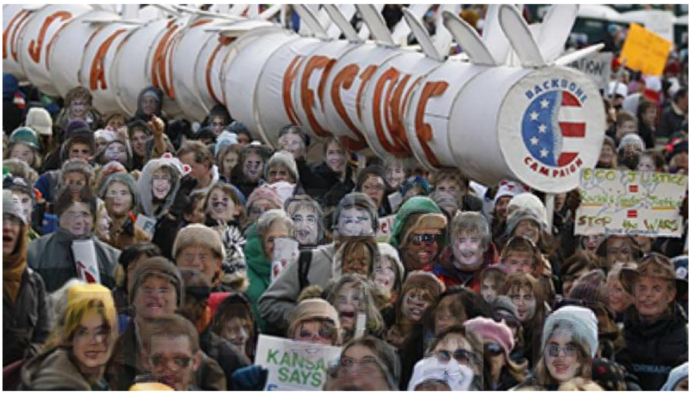


Figure 2: WIDER Faces example anonymized using new method

C Object detection

C.1 Annotate video

Create images from video using YOLO_Mark. Setting the last parameter to 0 instructs the program to save each video frame as an image. Setting to 1 will save every second frame etc.

```
yolo_mark.exe data/videoimg cap_video filename 0
```

Pseudo-labeling - to process a list of images train.txt and save results of detection in Yolo training format for each image as label ¡image_name¿.txt

```
darknet.exe detector test coco.data yolov3.cfg yolov3.weights
-dont_show -save\_labels < train.txt
```

C.2 Short Term Detection Memory

Modifications are implemented in Darknet fork.

```
typedef struct detectedObj {
    detection det;
    int missCount;
} detectedObj;

typedef struct detectionStore {
    detectedObj* store;
    int maxStoreCapacity;
    int storeLength;
} detectionStore;
```

where the Darknet defines

```
typedef struct box {
    float x, y, w, h;
} box;

typedef struct detection{
    box bbox;
    int classes;
    float *prob;
    float *mask;
    float objectness;
    int sort_class;
} detection;
```

C.3 Blurring anonymization filter

Blur a rectangle in the given image (mat). Use the rectangles width to determine the blur mask size.

```
void blurRectangle(cv::Mat* mat, cv::Point pt1, cv::Point pt2)
{
    int maskWidth = pt2.x - pt1.x;
    maskWidth = 2 * maskWidth / 3;
    //Ensure it has an odd number
    if (maskWidth % 2 == 0)
```

```
            maskWidth += 1;
        cv::Rect r(pt1, pt2);
        cv::Mat C = cv::Mat(*mat, r);

        cv::GaussianBlur(C, C, cv::Size(maskWidth, maskWidth), 0, 0, 4);
    }
```

UiA University of Agder
Master’s thesis
Faculty of Engineering and Science
Department of ICT

© 2020 Eivind Lindseth. All rights reserved